

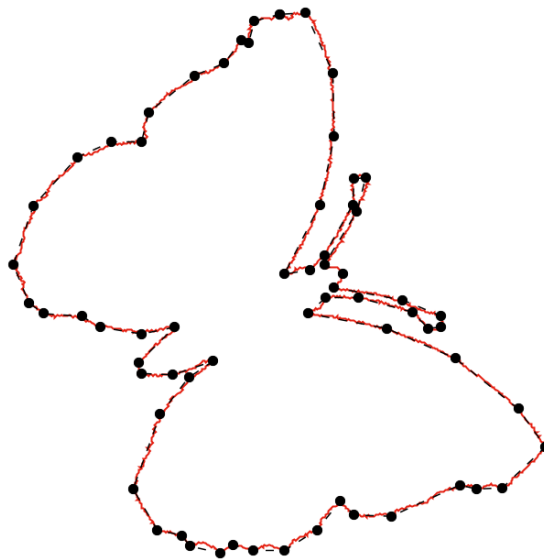


UNIVERSIDAD DE CÓRDOBA

Departamento de informática y análisis numérico

Tesis doctoral

*Contribución al reconocimiento de objetos 2D
mediante aproximaciones poligonales*



Luis del Moral Martínez

Córdoba, abril de 2017

TITULO: *CONTRIBUCIÓN AL RECONOCIMIENTO DE OBJETOS 2D
MEDIANTE APROXIMACIONES POLIGONALES*

AUTOR: *Luis del Moral Martínez*

© Edita: UCOPress. 2017
Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

www.uco.es/publicaciones
publicaciones@uco.es



UNIVERSIDAD DE CÓRDOBA

Departamento de informática y análisis numérico

Tesis doctoral

*Contribución al reconocimiento de objetos 2D
mediante aproximaciones poligonales*

Luis del Moral Martínez

Directores:

Dr. Nicolás Luis Fernández García

Dr. Rafael Medina Carnicer

Córdoba, abril de 2017



TÍTULO DE LA TESIS:

Contribución al reconocimiento de objetos 2D mediante aproximaciones poligonales

DOCTORANDO/A: Luis del Moral Martínez

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

Prof. Nicolás Luis Fernández García, Doctor en Informática, y Prof. Rafael Medina Carnicer, Doctor en Informática, ambos adscritos al Grupo de Investigación de “Aplicaciones de la Visión Artificial” de la Universidad de Córdoba y directores de la Tesis titulada “Contribución al reconocimiento de objetos 2D mediante aproximaciones poligonales” que ha sido desarrollada por D. Luis Del Moral Martínez

INFORMAN

Que la presente Tesis ha sido realizada como fruto de la investigación desarrollada en el grupo de investigación “Aplicaciones de Visión Artificial” y dentro del Programa de Doctorado “Ingeniería y Tecnología” de la Universidad de Córdoba, siguiendo los criterios de calidad exigidos a este tipo de trabajos.

El doctorando ha investigado en campo científico del Reconocimiento de Objetos 2D mediante el estudio y diseño de nuevos algoritmos de generación de aproximaciones poligonales. Como resultado de la investigación realizada, el doctorando contribuido al citado campo científico con la siguiente publicación en una revista indexada:

- N. L. Fernández García, L. del Moral Martínez, Á., F. J. Madrid Cuevas y R. Medina Carnicer. A new thresholding approach for automatic generation of polygonal approximations. *Journal of Visual Communication and Image Representation*, 2016, 35, páginas 155-168.

Por todo ello, los directores de la tesis creen que tanto el doctorando como su trabajo de investigación reúnen las condiciones exigidas a este tipo de trabajos y autorizan la presentación de la tesis doctoral

Córdoba, 19 de abril de 2017

Firma del/de los director/es

Fdo.: Nicolás Luis Fernández García

Fdo.: Rafael Medina Carnicer

Resumen

En la presente Tesis Doctoral se realizan aportaciones novedosas en las etapas de *descripción e interpretación* del proceso de reconocimiento de objetos bidimensionales.

Se proponen nuevas técnicas de umbralización unimodal aplicadas a la generación de aproximaciones poligonales. Estas técnicas se han comparado con las estrategias de umbralización clásicas propuestas por *Rosin*.

Se propone un nuevo método que obtiene aproximaciones poligonales de manera no supervisada; es decir, no paramétrica. Este método incorpora una etapa de *umbralización unimodal*.

Se ha realizado un análisis exhaustivo del método propuesto para diseñar nuevas versiones, según la combinación de las características de algunas de sus etapas. Se han considerado dos estrategias principales: estrategia de *división de puntos* y estrategia de *fusión* de puntos. Se han comparado las nuevas versiones propuestas con respecto al método original y se han obtenido nuevas versiones que representan una mejora considerable, mejorando también a todas las estrategias clásicas analizadas.

Se ha incorporado una fase final de optimización, que se basa en el método propuesto por *Masood*. Posteriormente, se ha realizado un estudio comparativo para seleccionar la versión más eficiente de cada estrategia, así como la versión que obtiene el mejor resultado de entre todas ellas. Las versiones optimizadas mejoran al algoritmo original propuesto y a todas las demás versiones analizadas.

En resumen, se ha desarrollado un nuevo método heurístico que permite generar aproximaciones poligonales eficientes de forma no supervisada. Este método puede ser utilizado en aplicaciones de tiempo real, superando las dificultades que presentan los algoritmos óptimos, que requieren de una carga computacional mayor.

Abstract

This doctoral thesis introduces original contributions to the *description* and *interpretation* stages of the bidimensional object recognition process.

A new unimodal thresholding approach has been proposed in order to generate polygonal approximations of bidimensional contours. These techniques have been compared with the classic thresholding techniques proposed by *Rosin*.

A new unsupervised method has been proposed. This method obtains polygonal approximations automatically and also includes the new unimodal thresholding approach proposed.

An exhausted analysis has been developed in order to design new versions of the proposed method, according to the combination of its different characteristics. Two strategies have been considered: point division (split) and point fusion (merge). All the versions have been compared with the original method and some of them prove to obtain a measurable improvement. Also, the new versions improve all the classic approaches that have been analyzed.

A final optimization stage has been incorporated. This optimization is based in the algorithm proposed by *Masood*. A comparative study has been developed and the best optimized method has been chosen. The optimized versions improve the original method proposed and all the versions analyzed.

To summarise, a new heuristic method has been developed. This approach can obtain efficient polygonal approximations automatically, uses a new unimodal thresholding algorithm and includes a final optimization stage. This method can be used in real time applications, exceeding the difficulties suffered by optimal algorithms, that need a higher computational load.

“La suerte favorece sólo a la mente preparada”.

Louis Pasteur.

*“No hay obstáculos imposibles;
sólo hay voluntades fuertes y débiles”.*

Julio Verne.

*“Lo supremo en el arte de la guerra consiste
en someter al enemigo sin darle batalla”.*

Tsun Zu.

*“Los dos días más importantes de tu vida son el día que
naces y el día que descubres porqué”.*

Mark Twain.

Agradecimientos

Hoy concluye uno de los proyectos más ambiciosos de mi vida. Ha sido un proceso gratificante, aunque en ocasiones bastante incierto. Este camino me ha enseñado el verdadero significado de la paciencia y la constancia, las dos compañeras sin las que habría sido imposible escribir hoy estas líneas.

Quiero comenzar dando las gracias a la Universidad de Córdoba y a los directores D. Nicolás Luis Fernández García y D. Rafael Medina Carnicer, profesores del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba, por haberme dado la posibilidad de emprender este trabajo, por haberse prestado a dirigirlo, por su esfuerzo y su amistad. Ha sido complicado compaginar esta labor investigadora con mi trabajo, pero siempre me han comprendido, apoyado y guiado.

A los profesores D. Ángel Carmona Poyato y D. Francisco José Madrid Cuevas, por haber puesto a mi disposición sus investigaciones.

A los profesores D. Juan Gómez Luna, D. José Manuel Palomares Muñoz, D. Joaquín Olivares Bueno, D. Ezequiel Herruzo Gómez, D. Manuel Agustín Ortíz López, D^a Amalia Zafra Gómez, D. Carlos García Martínez, D^a Cristina María Gámez Fernández, D^a Joaquina Berral Yerón, D^a Inmaculada Serrano Gómez, D. Miguel Ángel Gómez Nieto y D^a Irene Luque Ruíz, por sus enseñanzas y sus consejos.

A mis compañeros de trabajo y amigos, por interesarse en innumerables ocasiones y por preguntar sobre estos experimentos *sacados de una historia de ciencia ficción*.

A mi creciente familia, y en especial a mis padres, por todo el esfuerzo y sacrificio realizado a lo largo de estos años. Gracias por enseñarme el valor de la constancia.

Finalmente, quiero dar las gracias a María José, porque es el optimismo en persona. Gracias por evitar que me dé por vencido y por acompañarme durante el desarrollo de esta Tesis Doctoral.

Gracias a todos por enseñarme a luchar por mis sueños.

A mi familia

Índice general

Glosario de términos	v
1. Introducción	1
1.1. Visión por computador y reconocimiento de patrones	1
1.2. Reconocimiento de objetos bidimensionales	2
1.3. Sistemas de representación de contornos bidimensionales	3
1.3.1. Aproximaciones poligonales	5
1.4. Objetivos	6
1.5. Organización del documento	7
2. Antecedentes	9
2.1. Estrategias de generación de aproximaciones poligonales: <i>split</i> y <i>merge</i> . .	9
2.1.1. Estrategia de división de puntos: <i>split</i>	9
2.1.2. Estrategia de fusión de puntos: <i>merge</i>	12
2.2. Medidas de calidad	15
3. Nuevo método de obtención de aproximaciones poligonales	19
3.1. Introducción	19

3.2.	Trabajos relacionados	19
3.3.	Método propuesto para la generación de aproximaciones poligonales	21
3.3.1.	Primera etapa: selección de los puntos iniciales	23
3.3.2.	Segunda etapa: computación de los valores de significación de los puntos que no son iniciales	23
3.3.3.	Tercera etapa: computación de los valores de significación de los puntos iniciales	26
3.3.4.	Cuarta etapa: generación de la curva de significación normalizada	26
3.3.5.	Quinta etapa: búsqueda del umbral de los valores de significación normalizados	28
3.3.6.	Sexta etapa: umbralización de los valores de significación normalizados	31
3.4.	Evaluación del método y análisis de los resultados	32
3.4.1.	Primer experimento: selección del mejor método de umbralización de la quinta etapa del nuevo método	33
3.4.2.	Segundo experimento: comparación del método propuesto con los métodos no-paramétricos propuestos por Prasad	39
3.5.	Conclusiones	40
4.	Análisis exhaustivo del nuevo método de obtención de aproximaciones poligonales	49
4.1.	Introducción	49
4.2.	Alternativas basadas en la estrategia <i>split</i>	50
4.2.1.	Primera etapa: selección de puntos iniciales	50
4.2.2.	Segunda etapa: computación de los valores de significación de los puntos que no son iniciales	54
4.2.3.	Tercera etapa: computación de los valores de significación de los puntos iniciales	60
4.2.4.	Séptima etapa: borrado de puntos dominantes superfluos generados por el casco convexo (opcional)	60
4.3.	Alternativas basadas en la estrategia <i>merge</i>	61
4.3.1.	Primera etapa: selección de los puntos iniciales	63

4.3.2.	Segunda etapa: cálculo de la significación de los puntos del contorno	65
4.3.3.	Tercera etapa: cálculo de la significación acumulada	68
4.4.	Evaluación de los métodos y análisis de los resultados	69
4.4.1.	Primer experimento: selección de la mejor versión del método basado en la estrategia <i>split</i>	71
4.4.2.	Segundo experimento: selección de la mejor versión del método basado en la estrategia <i>merge</i>	75
4.5.	Conclusiones	81
5.	Optimización de los métodos basados en las estrategias <i>split</i> y <i>merge</i>	89
5.1.	Introducción	89
5.2.	Método de optimización	90
5.3.	Evaluación de los métodos y análisis de los resultados	93
5.3.1.	Primer experimento: selección de la mejor versión optimizada del método <i>split</i>	95
5.3.2.	Segundo experimento: selección de la mejor versión optimizada del método <i>merge</i>	105
5.4.	Conclusiones	107
6.	Conclusiones y futuros estudios	117
6.1.	Conclusiones	117
6.2.	Futuros estudios	118
A.	A new thresholding approach for automatic generation of polygonal approximations	119
	Bibliografía	135
	Índice de Figuras	143
	Índice de Tablas	147

Glosario de términos

Ajuste de la aproximación poligonal *Error de aproximación* al contorno de un objeto bidimensional.

Aproximación poligonal Polígono que se *ajusta* a un contorno de un objeto bidimensional.

Break point Punto en el que la línea que define un contorno *cambia de dirección*.

Casco convexo El polígono con *menor área* que envuelve a un contorno.

Centroide *Centro geométrico* de un objeto.

Contorno Conjunto de líneas que delimitan una *forma*.

Curva de significación Representación gráfica bidimensional de los *valores de significación* de un contorno.

Descriptor de la imagen Característica que representa a una imagen: *forma, color, textura*, etcétera.

Eje de mínima inercia Línea recta que *minimiza* las distancias cuadradas de los puntos del contorno.

Forma *Silueta* que se genera cuando un objeto bidimensional es iluminado por una fuente de luz situada a una distancia infinita.

Medida de calidad Magnitud que evalúa el ajuste de una *aproximación poligonal* a un contorno.

Punto dominante Punto que forma parte de la *aproximación poligonal* de un contorno.

Umbralización Efecto de seleccionar los puntos de un contorno utilizando un valor discriminante, denominado *umbral*, de forma que los valores superiores al umbral son seleccionados y el resto son descartados.

Valor de significación Grado de *importancia* de un punto.

1

Introducción

1.1. Visión por computador y reconocimiento de patrones

La *visión por computador* o *visión artificial* es una disciplina científica que desarrolla métodos que permiten emular los procesos de la visión humana en un ordenador.

El proceso que se emplea para reconocer objetos de una imagen del mundo real recibe el nombre de *reconocimiento de patrones*. Este proceso se engloba en un proceso que se denomina *procesamiento digital de imágenes*, que se divide a su vez en cinco etapas (véase la figura 1.1):

1. **Digitalización:** se captura una imagen del mundo real y se transforma en una *imagen digital* mediante un proceso de *discretización*. El proceso es susceptible a la presencia de ruido y a la pérdida de definición (que dependerá del dispositivo de captura empleado).
2. **Preprocesamiento:** su objetivo principal es transformar la *imagen digital* para obtener otra de mayor calidad que facilite el desarrollo de las siguientes etapas. Durante esta fase, se puede *eliminar o atenuar* la influencia del ruido presente en la imagen, *mejorar su contraste*, etc.
3. **Segmentación:** se *extrae información* de la *imagen digital* mediante su descomposición en *unidades de información* homogéneas.
4. **Representación y descripción:** se parametrizan los objetos identificados en la

etapa anterior y se extraen los *descriptores* de la imagen (elementos de información que permiten definir la imagen de forma cualitativa).

5. **Reconocimiento e interpretación:** se *reconocen* y *clasifican* cada uno de los objetos que forman la escena, en función de sus propiedades y características. Se interpretan los grupos de objetos reconocidos con la misión de dar un *significado* a la imagen.

El presente trabajo se centra en las dos últimas etapas de este proceso, que son las que aportan el significado a la imagen digitalizada.

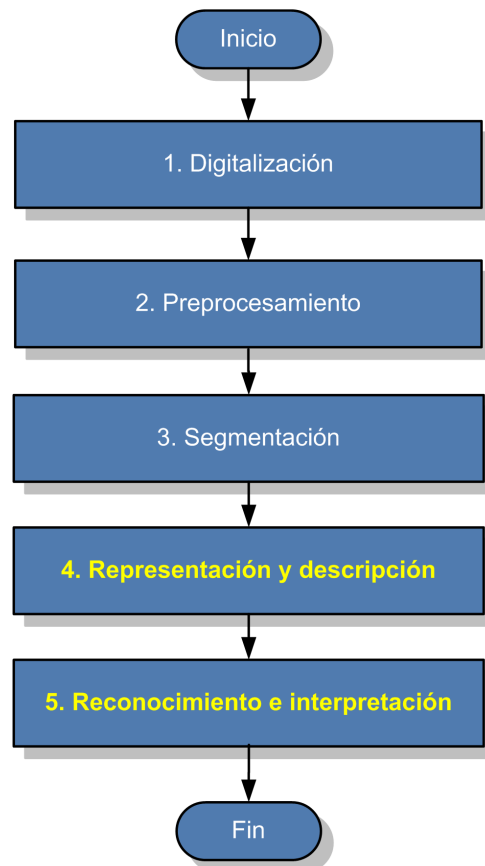


Figura 1.1: Fases del procesamiento digital de imágenes

1.2. Reconocimiento de objetos bidimensionales

Los algoritmos de reconocimiento de objetos permiten *identificar* o *clasificar* un objeto según sus *características* o *descriptores*: color, textura, forma, etc. El campo de estudio de este trabajo se centra en el reconocimiento de objetos mediante el análisis de su *forma*.

Existen diferentes áreas de conocimiento en las que es posible desarrollar aplicaciones basadas en el análisis y reconocimiento de formas bidimensionales [1]: el *reconocimiento de caracteres*, la *detección de tumores*, la *morfología celular*, etcétera.

Como consecuencia de esto, el análisis de formas bidimensionales ha sido un problema abordado por un amplio número de científicos de visión por artificial y reconocimiento de patrones [2–19].

Cuando el color y la textura no son características discriminantes, el reconocimiento de objetos puede usar la forma de los objetos (*shape analysis*). La *forma* de un objeto bidimensional es el *contorno* o silueta generada por un objeto cuando es iluminado por una fuente de luz situada a una distancia infinita [9, 20] (véase la figura 1.2).

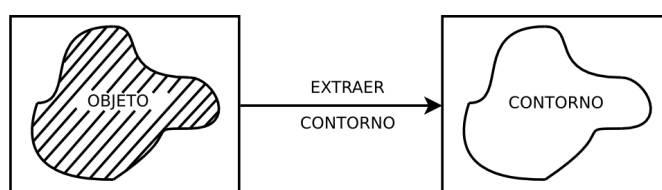


Figura 1.2: Extracción del contorno de un objeto de una imagen cualquiera

Generalmente, un método de análisis de formas bidimensionales consta de tres etapas: **representación**, que identifica los puntos que pertenecen a la forma; **descripción**, que caracteriza la forma obtenida a partir de su representación mediante un conjunto de descriptores, denominado *vector de características*; y, por último, **reconocimiento**, que clasifica un objeto a partir de la descripción realizada de la forma.

Los métodos de análisis de formas se clasifican en dos grandes grupos, según el tipo de puntos del objeto que utilizan:

- **Métodos basados en el contorno (métodos externos):** utilizan los puntos que forman parte del contorno (bordes) del objeto.
- **Métodos globales (métodos internos):** utilizan todos los puntos del objeto, tanto los pertenecientes al contorno como los interiores.

Este trabajo se centra en los métodos de análisis basados en el contorno o *externos*.

1.3. Sistemas de representación de contornos bidimensionales

Existen dos sistemas de representación de contornos bidimensionales:

- Representación matricial binaria.
- Sistemas de coordenadas.

Sin embargo, debido a la complejidad y al tamaño que pueden ocasionar estos sistemas de representación, es preciso recurrir a alternativas que permitan obtener una versión aproximada del contorno original. Esta aproximación se puede generar mediante el uso del *código de cadena* (véanse las figuras 1.3 y 1.4) o haciendo uso de una *aproximación poligonal*.

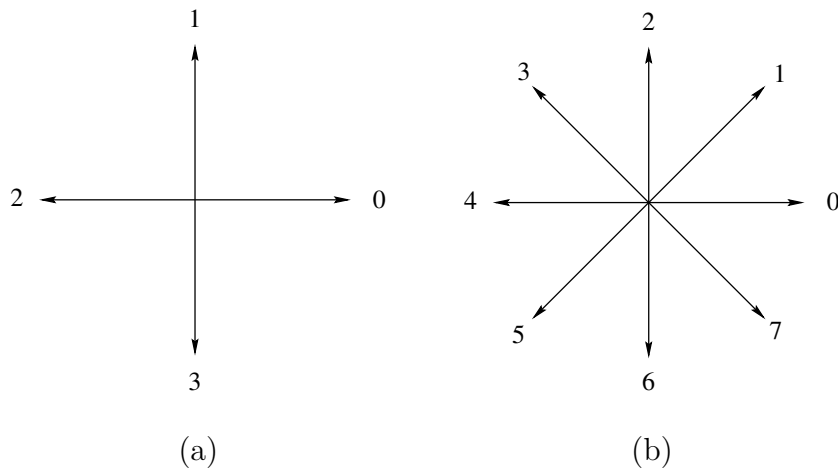


Figura 1.3: Direcciones de los códigos de cadena: (a) conectividad 4; (b) conectividad 8.

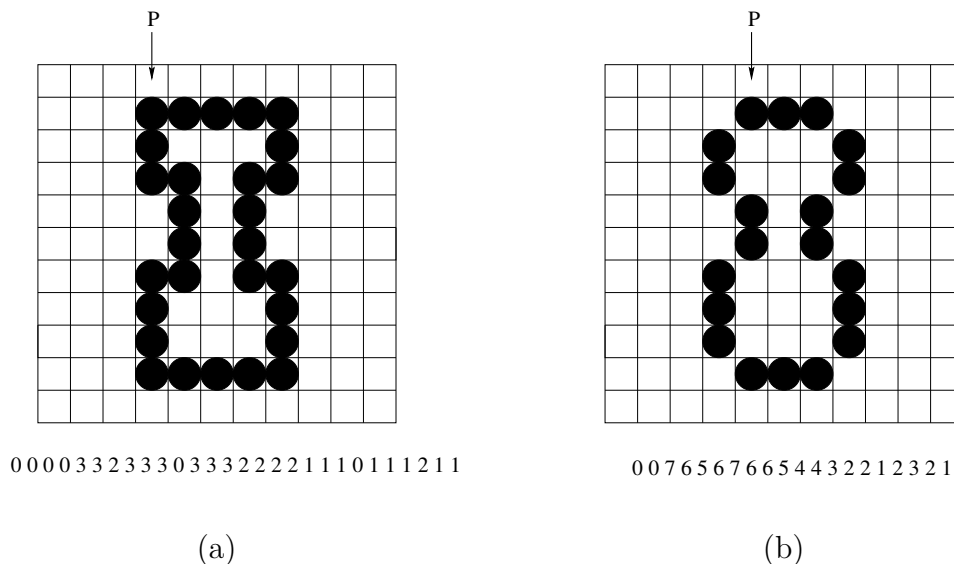


Figura 1.4: Códigos de cadena de (a) 4 direcciones y (b) 8 direcciones.

El inconveniente de estas técnicas radica en que el ajuste obtenido no se adapta en aquellos casos en los que el contorno presenta algún grado de curvatura. Para solucionar

este inconveniente, se han desarrollado diversos métodos que emplean las curvas *Splines* y *B-Splines* para disminuir la complejidad de la curva resultante [21–33].

El sistema de representación de contornos que se utiliza en este trabajo es el de las *aproximaciones poligonales*.

1.3.1. Aproximaciones poligonales

Existen algoritmos de reconocimiento que no precisan utilizar todos los puntos del contorno del objeto, sino aquellos que permiten definir las características esenciales del contorno. Las representaciones de contorno que se basan en este principio se denominan *aproximaciones del contorno* (*piecewise approximation*).

Una de las representaciones aproximadas más utilizadas es la *aproximación lineal o poligonal*, que genera un polígono cuyos vértices constituyen la representación del contorno original. Esta representación es muy popular debido a su simplicidad, localidad, generalidad y compresión [4, 9, 34–39]. Una aproximación poligonal será exacta si se utilizan todos los puntos del contorno como vértices del polígono. Sin embargo, el objetivo es conseguir la aproximación poligonal que contenga el menor número de vértices posible y conservando el mejor ajuste al contorno.

La aproximación poligonal de curvas planas cerradas (contornos), puede formularse de dos formas [40]:

- *Problema Min – ϵ (Minimum-distortion)*: estos métodos están basados en un número predefinido N de vértices. Intentan generar la aproximación poligonal *óptima* con N vértices de forma que su ajuste al contorno sea mínimo entre todas las aproximaciones poligonales con N vértices.
- *Problema Min – $\#$ (Minimum-rate)*: estos métodos se basan en una medida de error predefinida ϵ , e intentan generar la aproximación poligonal, con el mínimo número de vértices, cuyo ajuste al contorno sea menor que ϵ .

Se han propuesto numerosos métodos que permiten obtener aproximaciones poligonales de curvas planas cerradas o contornos. Estos métodos pueden clasificarse en dos categorías:

- *Métodos de optimización*: están basados en un criterio de optimización, como, por ejemplo, una función objetiva [41–43], un mínimo número de rectángulos de una anchura determinada [44], una curva cerrada cíclica de doble tamaño [40], la suma cuadrática de los errores de ajuste [45–47], la distancia de la ciudad de los bloques o de *Manhattan* [48, 49], etcétera.
- *Métodos heurísticos*: tratan de construir una aproximación poligonal utilizando características perceptuales o geométricas de los contornos [50]:

- *Métodos paramétricos*: tienen en cuenta uno o más parámetros para generar la aproximación poligonal. Algunos de estos métodos se basan en un parámetro de precisión seleccionado por el usuario [17, 18, 51–60], en un umbral de finalización [50], un nivel específico de significación [20], un segmento difuso máximo de anchura ν [61], en el filtro del espacio de escala [62–64] o en condiciones parametrizadas para la supresión de *break points* [41, 65], etcétera.
- *Métodos no paramétricos*: se han propuesto numerosos algoritmos que no utilizan parámetros para generar la aproximación poligonal de un contorno. Algunos de estos métodos se basan en los siguientes conceptos: región de soporte [66], filtro gaussiano y supresión de no-máximos [66], valor promedio del área y del coseno [67], código de la cadena incremental [68], fases de detección inicial y supresión [69], regiones de soporte y vectores de significación [20], código de concavidad [70, 71], selección de umbral [58], superposición de regiones de soporte [72], línea recta digital máxima [73], evolución del espacio de escalas [74], regiones de soporte y supresión de puntos colineales [19, 75], *framework* no paramétrico [76], región de soporte y coseno de k [77], cuasi-colinealidad [78], curvatura máxima global [79], unión de vértices significantes sucesivos [80], región de soporte y medida de la significación (por ejemplo, curvatura) [81], funciones de curvatura adaptativas [82] o región de soporte y curvatura máxima local [83].

Los métodos de optimización presentan dos inconvenientes [50]: la aproximación poligonal generada depende del criterio de optimización aplicado y requieren una alta complejidad computacional, que no es adecuada para aplicaciones de tiempo real. Por el contrario, los métodos heurísticos no garantizan ningún tipo de óptimo; sin embargo, estos algoritmos sí pueden generar aproximaciones poligonales aceptables para aplicaciones de tiempo real.

Las imágenes reales muestran formas irregulares y presentan mucha variabilidad y diferentes niveles de detalle. Solo un algoritmo que funcione independientemente de los parámetros de entrada, ajustándose a cada curva particular, podrá proporcionar resultados satisfactorios si se desea una representación eficiente y confiable del contorno original [55]. La necesidad de un algoritmo robusto y fiable que pueda ser aplicado sobre curvas diferentes de forma satisfactoria es un asunto de gran importancia [75].

En esta investigación se aborda el desarrollo de nuevos métodos basados *heurísticos* y automáticos. Es decir, *no paramétricos*.

1.4. Objetivos

El objetivo principal de esta Tesis Doctoral consiste en proponer nuevos métodos de generación de aproximaciones poligonales de contornos de objetos bidimensionales, teniendo en cuenta un enfoque *heurístico* y *no supervisado* y, por tanto, *no paramétrico*.

El enfoque heurístico debe estar basado en la *umbralización del nivel de significación* de los puntos del contorno para obtener los vértices de su aproximación poligonal.

1.5. Organización del documento

El resto del documento se ha estructurado en las siguientes partes:

- Revisión de los antecedentes relacionados con la obtención de aproximaciones poligonales de contornos bidimensionales (capítulo 2).
- Diseño de un nuevo método de generación de aproximaciones poligonales basado en la umbralización del nivel de significación de los puntos del contorno (capítulo 3).
- Análisis exhaustivo del método de generación de aproximaciones poligonales (capítulo 4).
- Incorporación y estudio de una etapa final de optimización al método propuesto (capítulo 5).
- Por último, el capítulo 6 resume las conclusiones del trabajo realizado y propone futuras líneas de investigación.

2

Antecedentes

2.1. Estrategias de generación de aproximaciones poligonales: *split* y *merge*

Existen diversos métodos de aproximaciones poligonales, según si se basan en el criterio de *división* de puntos del contorno (estrategia *split*), en el criterio de *fusión* de puntos del contorno (estrategia *merge*) o en una combinación de ambos. En este trabajo se utilizan los métodos *split* y *merge* por separado.

A continuación se describe en detalle cada una de las dos estrategias, haciendo referencia a las implementaciones del método *split* Ramer, Douglas–Peucker (RDP) y a los métodos *merge* de Carmona y Masood.

2.1.1. Estrategia de división de puntos: *split*

La estrategia *split* divide el contorno en sus partes más representativas mediante la elección de los puntos dominantes más significativos. Esta estrategia consta de las siguientes fases o etapas (véase la figura 2.1):

1. Buscar los puntos más extremos del contorno. Esto es, los que están situados más a la izquierda y más a la derecha, respectivamente.
2. Marcar los puntos extremos como dominantes.
3. Para cada par de puntos extremos (P_i, P_j) , repetir el siguiente proceso recursivo:

- a) Trazar la recta r que pasa por los puntos P_i y P_j .
- b) Calcular el punto P_k más distante a la recta r .
- c) Marcar el punto P_k como dominante.
- d) Repetir el proceso recursivo entre los puntos (P_i, P_k) y (P_k, P_j) .

El proceso finaliza cuando se satisface una condición de terminación. Esta condición puede estar relacionada con un número determinado de iteraciones o con un valor de ajuste de la aproximación final (si el método es paramétrico).

A continuación, se analiza la propuesta automatizada del método *split* de Ramer, Douglas–Peucker (RDP) [52, 84] presentada por Prasad et al. [76].

Método de Ramer, Douglas–Peucker (RDP) automatizado por Prasad

Se define $Q = \{P_1, \dots, P_n\}$ como el conjunto de *puntos* que componen el contorno original de un objeto bidimensional y $D = \emptyset$ como el conjunto de puntos dominantes estables de la aproximación poligonal. Entonces, se ejecuta el procedimiento *RDP*, que se compone de los siguientes pasos:

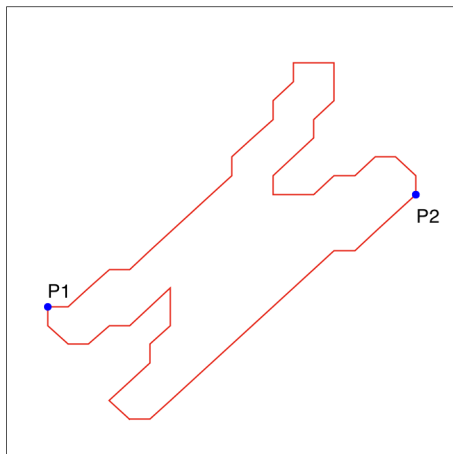
1. Buscar el punto P_i que esté situado *más a la izquierda* del contorno, marcarlo como *dominante* e introducirlo en el conjunto de puntos dominantes D .
2. Buscar el punto P_j que esté situado *más a la derecha* del contorno, marcarlo como *dominante* e introducirlo en el conjunto de puntos dominantes D .
3. Ejecutar el procedimiento recursivo ***RDP_recursivo(i, j)*** entre los puntos P_i y P_j (punto izquierdo - punto derecho):
 - a) Calcular la recta $r \equiv aX + bY + c = 0$ definida por los puntos P_i y P_j .
 - b) Para cada punto del contorno P_k , situado entre los puntos $P_{i+1} \in Q$ y $P_{j-1} \in Q$, calcular la distancia del punto a la recta r , según la expresión:

$$d(P_k, r) = \frac{|a x_k + b y_k + c|}{\sqrt{a^2 + b^2}} \quad (2.1)$$

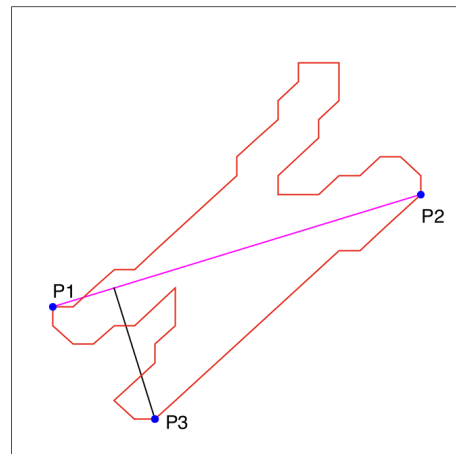
- c) Calcular la distancia máxima $d_{max}(P_k, r)$, $k \in \{i + 1, \dots, j - 1\}$, $P_k \in Q$.
- d) Calcular la distancia $s\delta\phi_{max}$, definida por la expresión:

$$s\delta\phi_{max} \approx \max \left(s \tan^{-1} \left\{ \frac{1}{s} (|\sin\phi \pm \cos\phi|) (1 - t_{max} + t_{max}^2) \right\} \right) \quad (2.2)$$

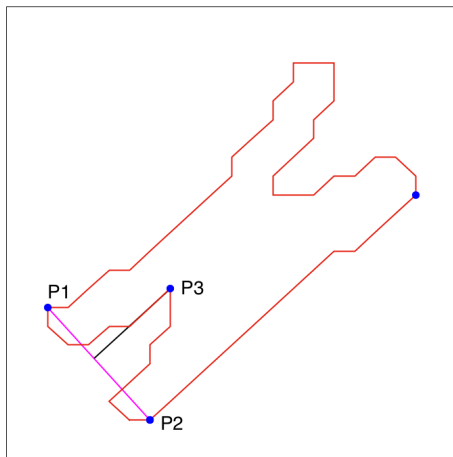
- e) Se comprueba si se cumple la condición $d_{max}(P_k, r) = s\delta\phi_{max}$:
 - Si $d_{max}(P_k, r) = s\delta\phi_{max}$, se termina el proceso recursivo.
 - Si $d_{max}(P_k, r) \neq s\delta\phi_{max}$, entonces se marca el punto P_k como dominante y se le introduce en el conjunto de puntos dominantes D . Se repite el proceso recursivo ***RDP_recursivo*** para las siguientes parejas de puntos:



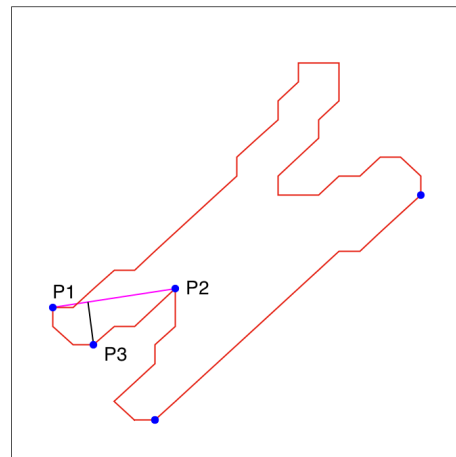
(a) Primera iteración



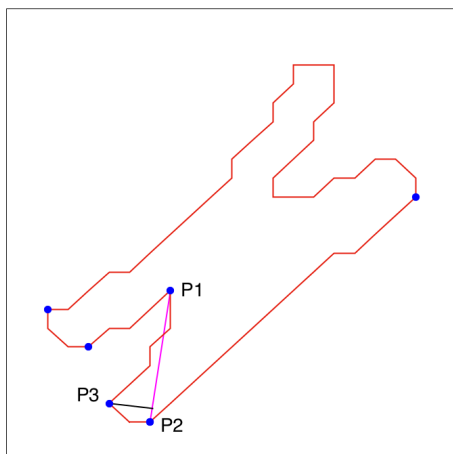
(b) Segunda iteración



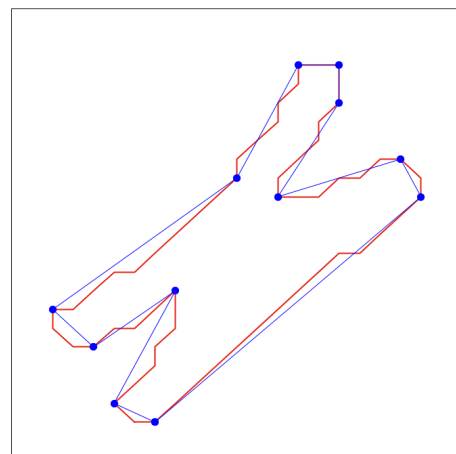
(c) Tercera iteración



(d) Cuarta iteración



(e) Quinta iteración



(f) Aproximación final (tras once iteraciones)

Figura 2.1: *Estrategia de división de puntos (split):* a lo largo del proceso, se divide el contorno en sus partes más representativas mediante la elección de los puntos dominantes más significativos. En cada iteración, se añade un nuevo punto dominante a la aproximación final.

- 1) ***RDP_recur_sivo*** (i, k) entre los puntos P_i y P_k .
 - 2) ***RDP_recur_sivo*** (k, j) entre los puntos P_k y P_j .
4. Ejecutar el procedimiento recursivo ***RDP_recur_sivo***(j, i) entre los puntos P_j y P_i (punto derecho - punto izquierdo):
- a) Calcular la recta $r \equiv aX + bY + c = 0$ definida por los puntos P_j y P_i .
 - b) Para cada punto del contorno P_k , situado entre los puntos $P_{j+1} \in Q$ y $P_{i-1} \in Q$, calcular la distancia del punto a la recta r , según 2.1.
 - c) Calcular la distancia máxima $d_{max}(P_k, r)$, $k \in \{j+1, \dots, i-1\}$, $P_k \in Q$.
 - d) Calcular la distancia $s\delta\phi_{max}$, definida en 2.2.
 - e) Se comprueba si se cumple la condición $d_{max}(P_k, r) = s\delta\phi_{max}$:
 - Si $d_{max}(P_k, r) = s\delta\phi_{max}$, se termina el proceso recursivo.
 - Si $d_{max}(P_k, r) \neq s\delta\phi_{max}$, entonces se marca el punto P_k como dominante y se le introduce en el conjunto de puntos dominantes D . Se repite el proceso recursivo ***RDP_recur_sivo*** para las siguientes parejas de puntos:
 - 1) ***RDP_recur_sivo*** (j, k) entre los puntos P_j y P_k .
 - 2) ***RDP_recur_sivo*** (k, i) entre los puntos P_k y P_i .

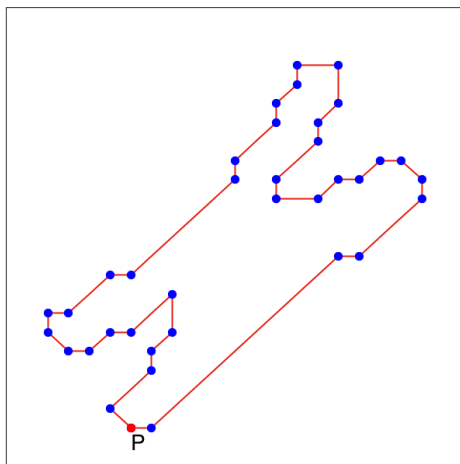
Al final del proceso, el conjunto D contiene todos los puntos dominantes que forman la aproximación poligonal del contorno bidimensional inicial.

2.1.2. Estrategia de fusión de puntos: *merge*

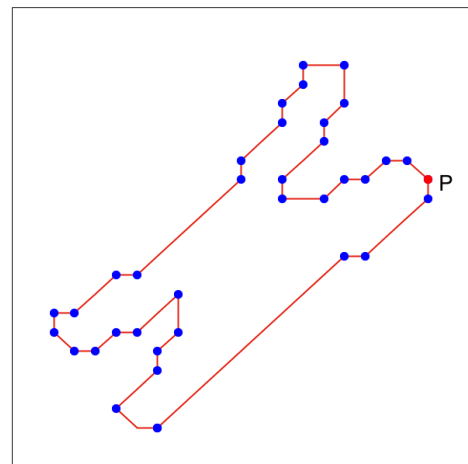
La estrategia *merge* parte de todos los puntos iniciales del contorno y los va suprimiendo (fusionando) uno a uno hasta obtener una aproximación poligonal. Esta estrategia consta de las siguientes fases o etapas (véase la figura 2.2):

1. Calcular el código de cadena del contorno (véase el anexo 2).
2. Marcar los *break points* como puntos dominantes.
3. Para cada par de puntos dominantes (P_i, P_j), repetir el siguiente proceso iterativo:
 - a) Trazar la recta r que pasa por los puntos P_i y P_j .
 - b) Para cada punto P_k del contorno original, medir la distancia d del punto P_k a la recta r .
 - c) Si la distancia d es menor que una tolerancia t fijada al inicio del algoritmo, entonces el punto P_k se marca como no dominante.

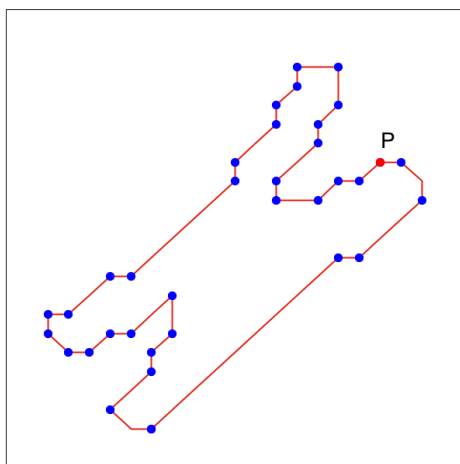
Inicialmente, esta estrategia parte de todos los puntos que forman el contorno y, tras sucesivas iteraciones, los fusiona hasta alcanzar cierto umbral de finalización o satisfacer una condición de terminación. A continuación se analizará la propuesta automatizada de los métodos de Carmona [53] y Masood [17, 57], presentadas por Prasad et al. [76].



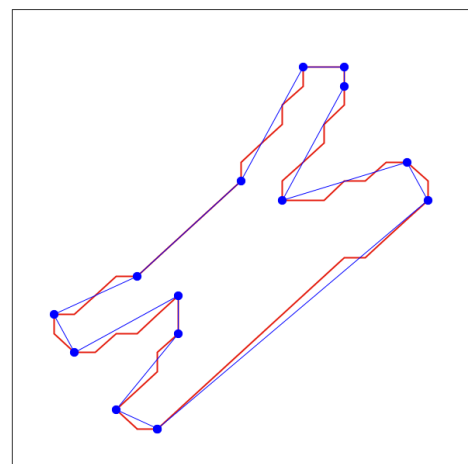
(a) Primera iteración



(b) Segunda iteración



(c) Tercera iteración



(d) Aproximación final (tras veinticuatro iteraciones)

Figura 2.2: Estrategia de fusión de puntos (*merge*): a lo largo del proceso, se fusionan los puntos del contorno hasta cumplir la condición de terminación.

Método de Carmona automatizado por Prasad

Se define $Q = \{Q_1, \dots, Q_n\}$ como el conjunto de *puntos* que componen el contorno original de un objeto bidimensional. Sobre este conjunto de puntos, se ejecuta el procedimiento *Carmona*, que se compone de los siguientes pasos:

1. Calcular el *código de cadena* de los puntos del contorno (véase el anexo 2).
2. Marcar los *break points* del contorno como puntos dominantes estables. Sea D el conjunto de puntos dominantes iniciales del contorno.
3. Sea d_{tol} el umbral empleado para discriminar los *puntos dominantes* de los *puntos no dominantes* del contorno.

Inicialmente, $d_{tol} \leftarrow 0.0$.

4. Repetir el siguiente proceso iterativo:

a) Obtener el punto dominante inicial $P_i \in D$, que será el punto que tenga una mayor distancia a la recta r , definida por los puntos dominantes anterior $P_{i-1} \in D$ y posterior $P_{i+1} \in D$.

b) Incrementar el valor del umbral d_{tol} .

$$d_{tol} \leftarrow d_{tol} + 0.5$$

c) Iterar para cada punto P_k in Q :

- Calcular los puntos dominantes anterior $P_{i-1} \in D$ y posterior $P_{i+1} \in D$.
- Calcular la recta $r \equiv aX + bY + c = 0$ definida por los puntos P_{i-1} y P_{i+1} .
- Calcular la distancia d del punto P_k a la recta r según la expresión 2.1.
- Si $d < d_{tol}$, entonces se marca el punto P_k como no dominante y se borra del conjunto D .

5. Comprobar la condición de parada del algoritmo:

- Para cada punto dominante $P_d \in D$:
 - Obtener el punto dominante posterior P_{d+1} .
 - Calcular la distancia d_n según la expresión:

$$d_n = \max \left(\frac{|x_m(y_m - y_{n+1}) + y_m(x_{n+1} - x_n) + y_{n+1}x_n - y_nx_{n+1}|}{\sqrt{(x_{n+1} - x_n)^2 + (y_n - y_{n+1})^2}} \right) \quad (2.3)$$

donde $m \in \{I_n, I_{n+1}, \dots, I_{n+1}\}$.

- Calcular la distancia d_n^{max} según la expresión:

$$d_n^{max} = S_n \phi_n \quad (2.4)$$

donde S_n representa la longitud del segmento de línea formado por el punto dominante P_d y el punto dominante posterior P_{d+1} y ϕ_n es el ángulo que forma el segmento de línea con el eje de abscisas.

- Si $d_n > d_n^{max}$, entonces se termina el método.

Método de Masood automatizado por Prasad

Se define $Q = \{Q_1, \dots, Q_n\}$ como el conjunto de *puntos* que componen el contorno original de un objeto bidimensional. Sobre este conjunto de puntos, se ejecuta el procedimiento *Masood*, que se compone de los siguientes pasos:

1. Calcular el *código de cadena* de los puntos del contorno (véase el anexo 2).
2. Marcar los *break points* del contorno como puntos dominantes estables. Sea D el conjunto de puntos dominantes iniciales del contorno.
3. Repetir el siguiente proceso iterativo:
 - a) Calcular el valor de error asociado (AEV o Associated Error Value):
 - Inicialmente, $AEV \leftarrow 0.0$.
 - Repetir para cada punto dominante $P_d \in D$:
 - 1) Calcular los puntos dominantes anterior $P_{d-1} \in D$ y posterior $P_{d+1} \in D$.
 - 2) Calcular la recta $r \equiv aX + bY + c = 0$ definida por los puntos P_{d-1} y P_{d+1} .
 - 3) Calcular la distancia d del punto P_k a la recta r según la expresión 2.1.
 - 4) Si $AEV < d$, entonces $AEV \leftarrow d$.
 - 5) Almacenar el valor AEV como la significación del punto P_d .
 - b) Calcular el punto $P_n \in D$ que posee el menor valor de AEV .
 - c) Calcular los puntos dominantes anterior $P_{n-1} \in D$ y posterior $P_{n+1} \in D$.
 - d) Calcular la distancia d_n^{max} según la expresión:

$$d_{n-1,n+1}^{max} = S_{n-1,n+1} \phi_{n-1,n+1} \quad (2.5)$$

- e) Si la significación del punto mínimo P_n es igual a $d_{n-1,n+1}^{max}$, se termina el método. En el caso contrario, continuar en el punto f).
- f) Si la significación del punto mínimo P_n no es igual a $d_{n-1,n+1}^{max}$, entonces se marca el punto como no dominante y se estabiliza el contorno según el método propuesto por Prasad.

2.2. Medidas de calidad

La *calidad* de una aproximación poligonal se puede evaluar de dos formas:

1. **Evaluación visual subjetiva:** un observador humano compara visualmente la aproximación poligonal con el contorno original.

- **Ventajas:**
 - Es un proceso fácil de aplicar.
- **Inconvenientes:**
 - El proceso no es automatizable.
 - El criterio de dos observadores podría diferir ante la misma aproximación poligonal.

2. Evaluación objetiva:

- Este proceso tiene en cuenta los siguientes criterios:
 - a) Número de puntos de la aproximación poligonal (*ratio de compresión*). El objetivo es aumentar la *ratio de compresión*. Es decir, obtener una aproximación poligonal con el menor número de puntos posible pero que se ajuste al contorno adecuadamente.
 - b) Error de ajuste de la aproximación poligonal al contorno. Este criterio mide la discrepancia ente el contorno y la aproximación poligonal. El objetivo es minimizar el error de ajuste.
- **Ventajas:**
 - El proceso es automatizable.
- **Inconvenientes:**
 - Los criterios en los que se basa son *contrapuestos*. En general, a medida que se reduce el *número de puntos* de la aproximación, se mejora la *ratio de compresión* pero se reduce la calidad del *ajuste*. Por el contrario, a medida que el *número de puntos* de la aproximación se incrementa, la *ratio de compresión* disminuye y la calidad del *ajuste* aumenta.
- El *objetivo* de la evaluación objetiva es lograr un **equilibrio** entre el *número de puntos* y el *ajuste* de la aproximación poligonal.

Para evaluar de manera objetiva la calidad de las aproximaciones poligonales, se han propuesto diversas medidas de calidad [42, 53, 85–88]:

- Para evaluar el primer criterio, es decir, el número de puntos de la aproximación poligonal, se utiliza la medida denominada *ratio de compresión* (cuyo acrónimo es CR, que proviene del término anglosajón *compression ratio*). Esta medida se define como el cociente entre el número de puntos del contorno n y el número de puntos que forman parte de la aproximación poligonal n_d (número de puntos dominantes). Cuanto menor sea el número de puntos dominantes del contorno n_d , mayor será la *ratio de compresión* (véase la expresión 2.6).

$$CR = \frac{n}{n_d} \quad (2.6)$$

- Para evaluar el segundo criterio, es decir, el error de ajuste de la aproximación poligonal al contorno, se utilizan dos medidas diferentes:
 - *Error cuadrático integral* (cuyo acrónimo es *ISE*, que proviene del término anglosajón *integral square error*): se calcula como la suma del cuadrado de la distancia de cada punto Q del contorno al lado de la aproximación poligonal definido por los puntos dominantes P_i y P_{i+1} . Véanse la expresión 2.7 y la figura 2.3.

$$ISE = \sum_{i=1}^n e_i^2 \quad (2.7)$$

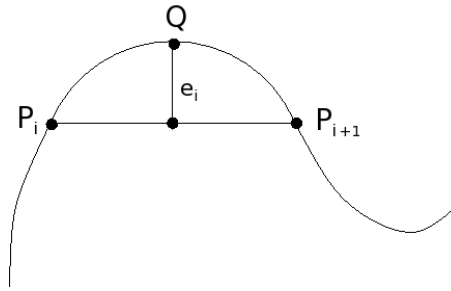


Figura 2.3: Distancia de un punto Q al lado de la aproximación poligonal definido por los puntos P_i y P_{i+1}

- *Error máximo* $E_\infty = \max_{i \in \{1, \dots, n\}} \{e_i\}$: mide la distancia máxima de los puntos del contorno a la aproximación poligonal. En ciertos casos, la medida *ISE* puede compensar un gran error de desviación de la aproximación poligonal al contorno, debido a que tenga un mejor ajuste en otras partes de la curva. La medida *Error máximo* pretende evitar este inconveniente [17].

Con el objetivo de combinar el *número de puntos* y el *ajuste* de la aproximación poligonal, se han propuesto algunas medidas adicionales:

- *Figura de Mérito* (o *Figure of Merit*): Sarkar [80] propuso la medida $FOM = \frac{CR}{ISE}$ para compensar la ratio de compresión (*CR*) y el error de ajuste (*ISE*) [18]. Rosin [88] demostró que los dos términos *CR* y *ISE* que se utilizan en la medida *FOM* no están balanceados, lo que provoca que esta medida tienda hacia aproximaciones con un valor inferior de la medida *ISE* (que puede ser fácilmente conseguido si se aumenta el número de puntos dominantes detectados). Este inconveniente se hace más evidente en contornos reales, que usualmente contienen un gran número de puntos. Por tanto, la medida *FOM* no es la mejor para realizar comparaciones de contornos con diferente número de puntos dominantes.
- *Suma ponderada del error cuadrático* (o *Weighted Sum of Squared Error*): la medida $WE = \frac{ISE}{CR}$ se define como la inversa de la medida *FOM* [65, 83].

- *Error máximo ponderado* (o *Weighted Maximum Error*): la medida $WE_\infty = \frac{E_\infty}{CR}$ también se ha propuesto para medir el rendimiento de las aproximaciones poligonales [17, 56, 83]. Técnicamente, las medidas WE y WE_∞ son similares a la medida FOM y sufren los mismos problemas [17].
- *Suma ponderada del error cuadrático parametrizada*: también se ha propuesto una versión parametrizada de la medida WE y que se define como $WE_n = \frac{ISE}{(CR)^n}$ [19, 42, 53, 61]. Esta medida permite balancear la contribución de las medidas ISE y CR , donde n puede tomar los valores $n = 1, 2, 3$. Carmona [41, 53] demostró que el valor $n = 2$ obtiene el mejor rendimiento de entre las tres opciones disponibles y, por tanto, este será el valor utilizado en los experimentos llevados a cabo en este trabajo.

Para evitar el inconveniente de la falta de balanceo de la medida FOM , Rosin [88] propuso una nueva medida denominada $merit = \sqrt{fidelity \times efficiency}$, donde $fidelity = \frac{E_{opt}}{E_{appr}} \times 100$ y $efficiency = \frac{N_{opt}}{N_{appr}} \times 100$, siendo E_{appr} y N_{appr} el error de ajuste y el número de puntos dominantes, respectivamente, de la aproximación poligonal que se desea evaluar. E_{opt} es el error de la aproximación poligonal óptima con el mismo número de puntos dominantes N_{appr} . Por último, N_{opt} representa el número de puntos dominantes que serían necesarios en la aproximación poligonal óptima para producir el mismo error E_{appr} [45].

El término *fidelity* mide cómo de bueno es el ajuste de la aproximación poligonal a la curva original en relación con el polígono óptimo, en términos de error de aproximación. El término *efficiency* mide la compacidad del polígono obtenido por el algoritmo que va a ser evaluado en relación con el polígono óptimo que incurre en el mismo error [88].

La medida de evaluación de Rosin puede comparar resultados de diferentes algoritmos con diferente número de puntos dominantes. Sin embargo, Masood [18] estableció que este método también sufre de algunos inconvenientes. Por ejemplo, un polígono que únicamente esté formado por *break points* producirá unos valores de $fidelity = 100$, $efficiency = 100$ y $merit = 100$. Esto significa que el conjunto de *break points* tomados como puntos dominantes produciría una aproximación poligonal perfecta, pero dicha aproximación no tendría ningún uso práctico, puesto que la ratio de compresión sería demasiado baja. Carmona [85] propuso una nueva medida para evaluar la calidad de las aproximaciones poligonales de las curvas. Esta medida está basada en una función objetivo que trata de minimizar la medida WE_2 utilizando las aproximaciones poligonales óptimas generadas por el algoritmo de Pérez y Vidal [45]. El principal inconveniente de estas medidas propuestas por Rosin [88] y Carmona [85] es que requieren soluciones óptimas, que son, en términos de computación, muy costosas.

En resumen, cuando el número de puntos dominantes DP es el mismo, las medidas ISE y E_∞ son las medidas cuantitativas más apropiadas para la comparación de aproximaciones poligonales [17]. Ahora bien, si el número de puntos dominantes DP es diferente, entonces la medida parametrizada WE_2 demuestra un mejor rendimiento [41].

3

Nuevo método de obtención de aproximaciones poligonales

3.1. Introducción

Este capítulo describe un nuevo método para la generación automática de aproximaciones poligonales de contornos 2D cerrados. El método que se propone computa en primer lugar el nivel de significación de los puntos del contorno usando una nueva versión simétrica del conocido método de Ramer, Douglas–Peucker [52, 84]. En un segundo paso, una nueva propuesta *Adaptativa* es aplicada para umbralizar el nivel de significación normalizado de los puntos del contorno y generar la aproximación poligonal. Los experimentos han mostrado que el método propuesto genera de forma eficiente las aproximaciones poligonales de contornos 2D cerrados. Se debe resaltar que la propuesta que se hace no requiere que ningún parámetro deba ser ajustado, por lo que puede considerarse como un método no supervisado y, por tanto, automático.

La organización del capítulo es la siguiente: la sección 3.2 describe los trabajos relacionados; la sección 3.3 presenta el nuevo método que se propone; los experimentos realizados y los resultados obtenidos son descritos en la sección 3.4 y, por último, la sección 3.5 resume las principales conclusiones obtenidas.

3.2. Trabajos relacionados

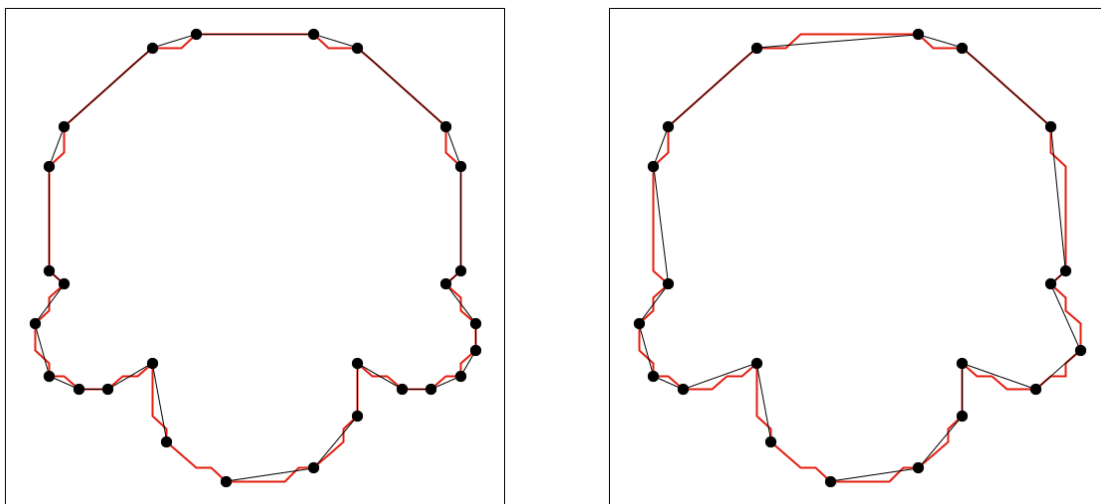
El primer paso del método propuesto se basa en una modificación del método Ramer, Douglas-Peucker (RDP) [52, 84], que utiliza una estrategia de división (*split*) independiente de la escala para obtener las partes más significativas de un contorno. Para aplicar el

método RDP y generar la aproximación poligonal de un contorno cerrado, se aplica el siguiente proceso:

1. Encontrar los puntos dominantes *superior-izquierdo* e *inferior-derecho* del contorno. Estos puntos se considerarán como los *puntos dominantes iniciales*.
2. Calcular el *punto más alejado* del segmento definido por dos puntos dominantes consecutivos [89]. El punto seleccionado se considera un nuevo punto dominante.
3. Repetir el paso 2 hasta que el conjunto de puntos dominantes genere una aproximación poligonal que se ajuste al contorno con un error predefinido al inicio del método.

El método RDP presenta dos inconvenientes (véase la figura 3.1):

1. Es un método *paramétrico*, puesto que el número de puntos dominantes finales (DP) depende de un error prefijado al inicio del método.
2. La *simetría* del contorno no se tiene en cuenta a la hora de generar la aproximación poligonal.



(a) Error predeterminado: 0.7 (DP = 27)

(b) Error predeterminado: 0.9 (DP = 20)

Figura 3.1: Inconvenientes del método RDP original: dependencia de un error paramétrico y pérdida de simetría. La figura muestra las aproximaciones poligonales generadas del contorno *semicircles* con dos errores predeterminados y el número de puntos dominantes de la aproximación final (DP).

3.3. Método propuesto para la generación de aproximaciones poligonales

El método que se propone para generar las aproximaciones poligonales consta de seis pasos (véase la figura 3.2):

1. *Selección de los puntos iniciales*: se consideran algunos puntos **especiales** del contorno para utilizarlos como *puntos iniciales*. Véase la sección 3.3.1.
2. *Cálculo de la significación de los puntos que no son iniciales*: se aplicará una *nueva* versión del método RDP [52, 84], que tiene en cuenta la simetría del contorno, para calcular los valores de significación de los puntos que no son iniciales. Esta nueva versión simétrica se describe en la sección 3.3.2.
3. *Cálculo de la significación de los puntos iniciales*: los valores de la significación de los *puntos iniciales* deberán ser mayores que los valores de significación de los otros puntos del contorno; debido a ello, en la sección 3.3.3 se propone un mecanismo para garantizarlo.
4. *Generación de la curva de significación normalizada*: los valores de significación de los puntos de contorno son utilizados para generar una *curva de significación*, que deberá ser normalizada para facilitar la búsqueda de los puntos dominantes mediante el uso de un método de umbralización. Este proceso es descrito en la sección 3.3.4.
5. *Búsqueda del umbral de los valores de significación normalizados*: la curva de significación normalizada será utilizada para buscar el umbral de los valores de significación normalizados. Para este propósito se han ensayado seis métodos de umbralización diferentes, los cuales se describen en la sección 3.3.5. El rendimiento de estos métodos ha sido analizado en un primer experimento (véase la sección 3.4.1) y se ha concluido que el método que se denomina *adaptativo*, que es una nueva propuesta de umbralización, proporciona los mejores resultados. Como consecuencia de ello, ha sido el método escogido para esta etapa.
6. *Umbralización de los valores de significación normalizados para obtener los puntos dominantes*: los valores de significación normalizados de los puntos del contorno serán umbralizados para obtener los puntos dominantes. Los puntos del contorno cuyo valor de significación normalizado sea mayor que el umbral serán seleccionados como *puntos dominantes* (véase la sección 3.3.6).

En el mejor de los casos, la complejidad computacional será $\mathcal{O}(n \log(n))$; en cambio, en el peor de los casos, si todos los break points son dominantes, la complejidad computacional será $\mathcal{O}(n^2)$ (véase el documento [90], en el que se analiza el método RDP y su complejidad computacional). La figura 3.2 muestra un esquema general del método que se propone.

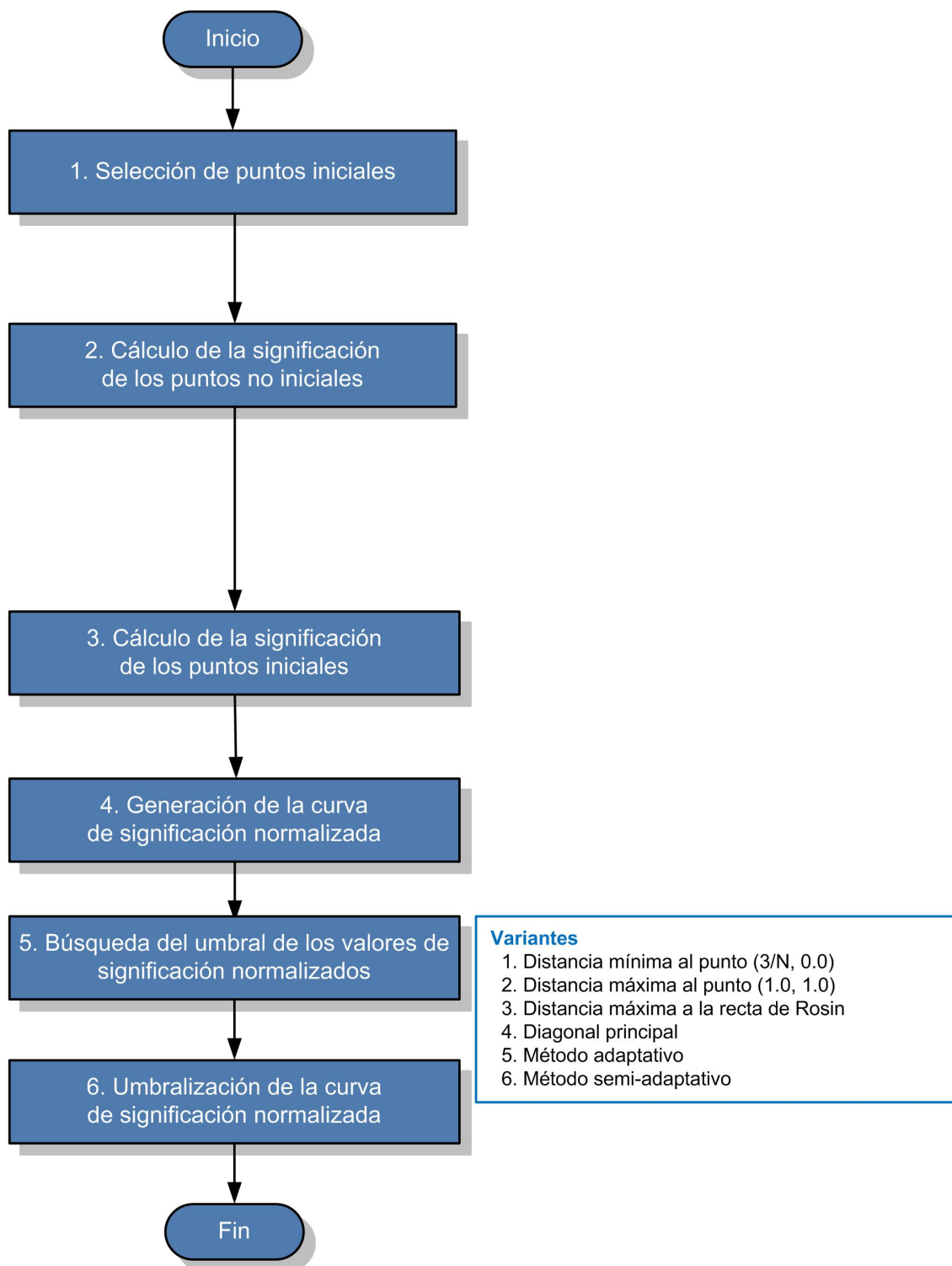


Figura 3.2: Esquema del método *split* propuesto

3.3.1. Primera etapa: selección de los puntos iniciales

El método utilizado para seleccionar los *puntos iniciales* de la aproximación poligonal consta de dos fases (véase la figura 3.3):

1. Seleccionar como *puntos iniciales* a los puntos más alejados del centroide.
2. Además, considerar también como *puntos iniciales* a los puntos más alejados de los puntos obtenidos en la fase anterior.

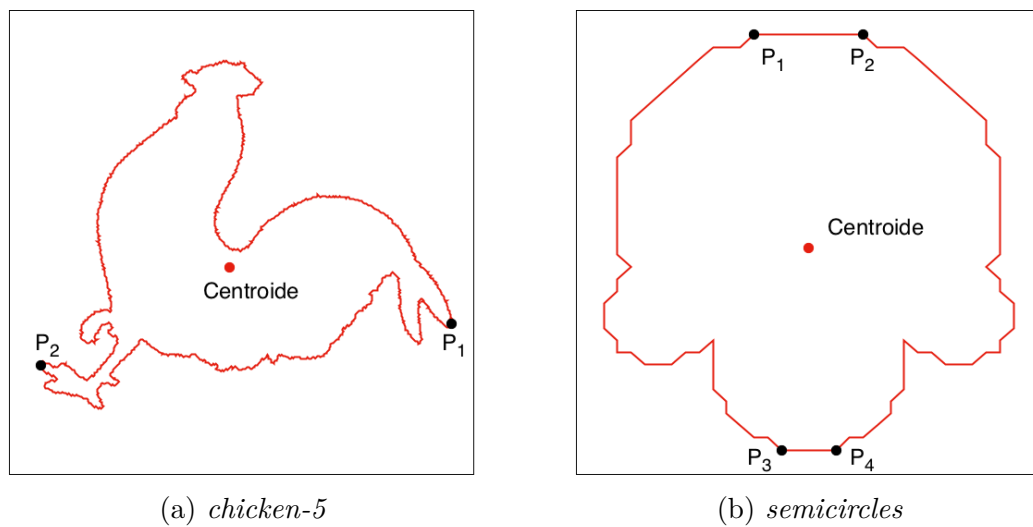


Figura 3.3: Primera etapa del método *split*. Selección de los puntos *iniciales* del contorno: (a) P_1 es el punto más alejado del centroide del contorno y el punto P_2 es el punto más alejado del punto P_1 ; (b) P_1 y P_2 son los puntos más alejados del centroide del contorno y P_3 y P_4 son los puntos más alejados de los puntos anteriores, respectivamente.

Este método es invariante a rotaciones y cambios de escala y no depende de la elección del primer punto en la secuenciación de los puntos del contorno. Además, tiene en cuenta la simetría del contorno debido al uso del centroide como medio para obtener los puntos iniciales. La complejidad computacional de esta etapa es lineal $\mathcal{O}(n)$, puesto que se recorre el contorno una única vez para buscar aquellos puntos que están más alejados del centroide.

3.3.2. Segunda etapa: computación de los valores de significación de los puntos que no son iniciales

El método propuesto para calcular la significación de los puntos que *no son iniciales* modifica el método RDP original [52, 84] para tener en cuenta la simetría del contorno.

Se define $I = \{P_1, \dots, P_n\}$ como el conjunto de *puntos iniciales* obtenido de acuerdo a lo indicado en la sección 3.3.1. Entonces, se ejecuta un procedimiento recursivo que denominamos *calcular_significación*. El procedimiento $\text{calcular_significación}(P_i, P_{i+1})$ debe ejecutarse para cada pareja de puntos iniciales consecutivos P_i y $P_{(i+1) \bmod n}$ ($\forall i \in \{1, \dots, n\}$) y consta de los siguientes pasos:

1. Calcular la distancia máxima d desde el segmento definido por P y P' a los puntos del contorno que se encuentran entre P y P' .
2. Si la distancia d es igual a 0.0, se asigna el valor de significación 0.0 a cada punto del contorno que se ubique entre los puntos P y P' . En este caso, el proceso recursivo se detiene.
3. Por el contrario, si la distancia d no es igual a 0.0 entonces:
 - a) Seleccionar los puntos del contorno comprendidos entre P y P' que estén más alejados del segmento definido por P y P' . Nótese que:
 - En la mayoría de los casos, solo existirá un punto, y el proceso continuará en el paso 2.
 - Existe la posibilidad de que exista más de un punto (por ejemplo, si el contorno tiene un lado recto que es paralelo al segmento definido por los puntos P y P'). En esta situación, solamente se seleccionan, entre ellos, los puntos que están más alejados del centroide del contorno. Estos puntos serán considerados como *puntos candidatos* (véase la figura 3.5).
 - Este proceso tiene en cuenta la simetría del contorno.
 - b) Siendo Q_1, Q_2, \dots, Q_m los *puntos candidatos* seleccionados en el apartado anterior:
 - Asignar la distancia máxima d como la *significación* de cada punto Q_i , $i \in \{1, \dots, m\}$.
 - Almacenar el valor máximo de significación en la variable Sig_{max} , que será utilizada en etapas posteriores del método.
 - Ejecutar el proceso recursivo $\text{calcular_significación}(P, P')$ con cada par de puntos P y Q_1 , Q_1 y Q_2 , \dots , Q_{m-1} y Q_m y, finalmente, Q_m y P' .

En resumen, el valor de significación de cada punto *no inicial* es su error de desviación d (véase la figura 3.4), teniendo en cuenta la simetría del contorno.

La complejidad computacional de esta etapa es $\mathcal{O}(n \log(n))$ en el mejor de los casos y $\mathcal{O}(n^2)$ en el peor, si todos los break points son dominantes (véase el documento [90], en el que se analiza el método RDP y su complejidad computacional).

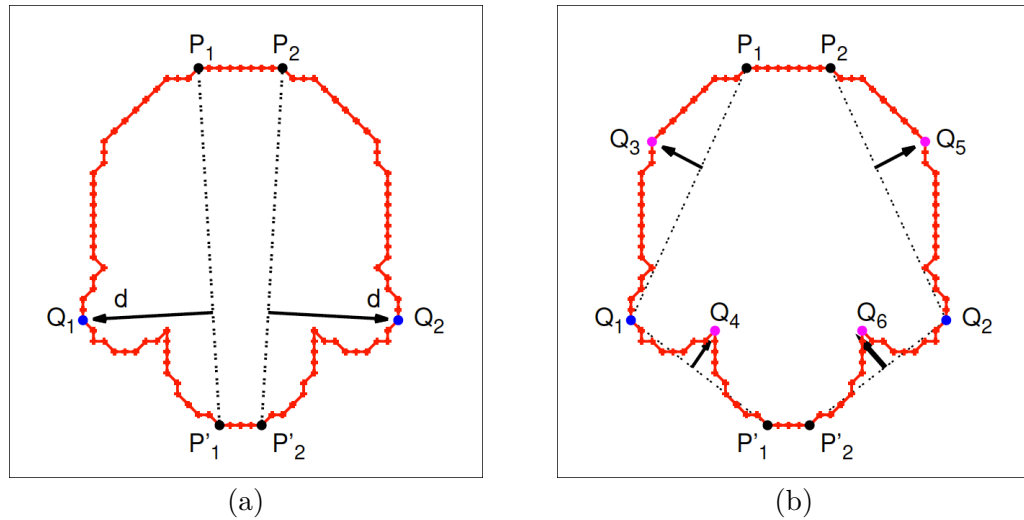


Figura 3.4: Segunda etapa del método *split*. Computación de los valores de significación de los puntos que no son iniciales: desarrollo del proceso recursivo *calcular_significación*(P, P') en el contorno *semicircles*: (a) Primeros 2 puntos candidatos Q_1 y Q_2 y su valor de significación d ; (b) siguientes 4 puntos candidatos: Q_3 - Q_6 .

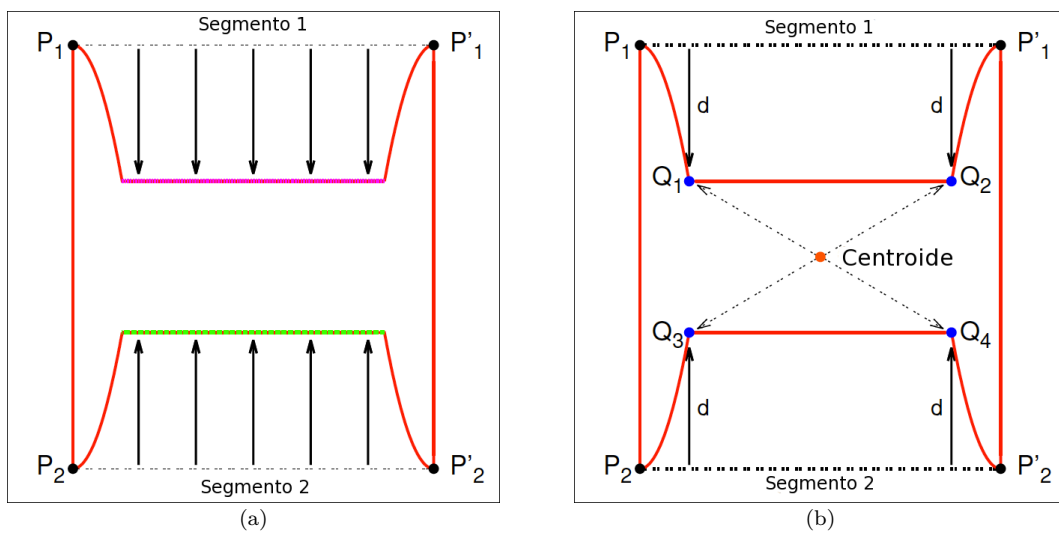


Figura 3.5: Segunda etapa del método del método *split*. Computación de los valores de significación de los puntos que no son iniciales: contorno con lados rectos. (a) puntos más alejados de los segmentos definidos por los *puntos iniciales* P_1, P'_1 y P_2, P'_2 ; (b) puntos más alejados del centroide respecto a los puntos anteriores: d es el valor de significación de los puntos candidatos Q_1 - Q_4 .

3.3.3. Tercera etapa: computación de los valores de significación de los puntos iniciales

El propósito de esta etapa es garantizar que los puntos iniciales tengan el valor máximo de significación. Inicialmente, el valor *máximo* de significación de los puntos que *no son iniciales* se computa como Sig_{max} (valor obtenido en la etapa anterior del método, descrita en la sección 3.3.2):

- Si Sig_{max} es igual a 0.0, entonces el valor de significación de los puntos *iniciales* será 1.0. Esta situación ocurre con contornos artificiales en los que los puntos iniciales definen una aproximación poligonal exacta.
- Por el contrario, si Sig_{max} no es igual a 0.0 entonces el valor de significación de los puntos *iniciales* será el máximo entre Sig_{max} y d , donde d es la distancia máxima desde el *centroide* hasta los puntos del contorno (véase la figura 3.6).

La complejidad computacional de esta etapa es lineal $\mathcal{O}(n)$, puesto que el cálculo del valor máximo de significación de los puntos del contorno se realiza mediante un proceso lineal. La distancia máxima de los puntos del contorno al centroide ya se ha calculado en la primera etapa del método.

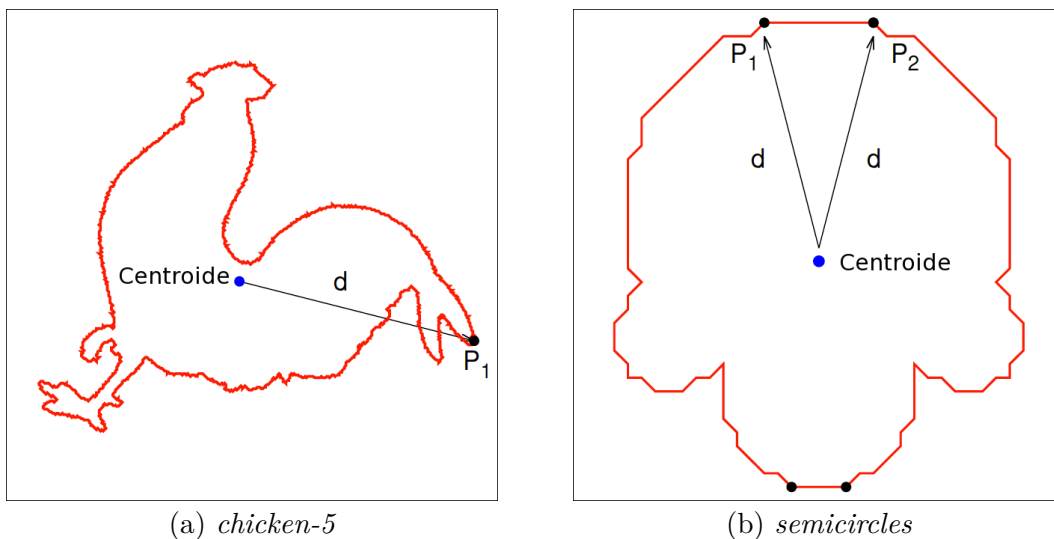


Figura 3.6: Tercera etapa del método *split*. Computación de los valores de significación de los puntos iniciales: d es la distancia máxima desde el centroide a los puntos del contorno.

3.3.4. Cuarta etapa: generación de la curva de significación normalizada

La *curva de significación normalizada* es una gráfica que representa el error de ajuste o desviación de la aproximación poligonal respecto del contorno y que se define en

función del número de puntos dominantes (véase la figura 3.7). Esta curva es estrictamente decreciente con una pendiente pronunciada. En el principio de la curva, la reducción del error es significativa; sin embargo, a medida que se van seleccionado cada vez más puntos dominantes, esta reducción decrece a medida que se aproximan los pequeños detalles del contorno. La normalización de la curva de significación intenta equiparar la influencia del número de puntos y los valores de la significación. El proceso de normalización consta de tres pasos:

1. Los puntos del contorno se ordenan en orden decreciente, según sus valores de significación.
2. Se normalizan los valores de significación en el rango $[0.0, 1.0]$. Este proceso de normalización se lleva a cabo realizando la división del valor de significación entre el valor de significación máximo.
3. Se normaliza el número de puntos del contorno en el rango $[0.0, 1.0]$. Este proceso de normalización se realiza por medio de la división del *índice* del punto del contorno entre el número N , que dicta el número total de puntos que tiene el contorno.

Esta curva de significación normalizada es empleada en la quinta etapa del método (véase la siguiente sección 3.3.5) para localizar el *punto de corte* o *esquina* (umbral), en el que el comportamiento de la curva cambia desde la aproximación de las partes importantes del contorno a la aproximación de los detalles más sutiles del mismo.

La complejidad computacional de esta etapa es $\mathcal{O}(n \log(n))$, puesto que el proceso de normalización es un proceso lineal $\mathcal{O}(n)$ y los valores de significación normalizada se pueden ordenar por medio de un método cuya complejidad computacional sea $\mathcal{O}(n \log(n))$.

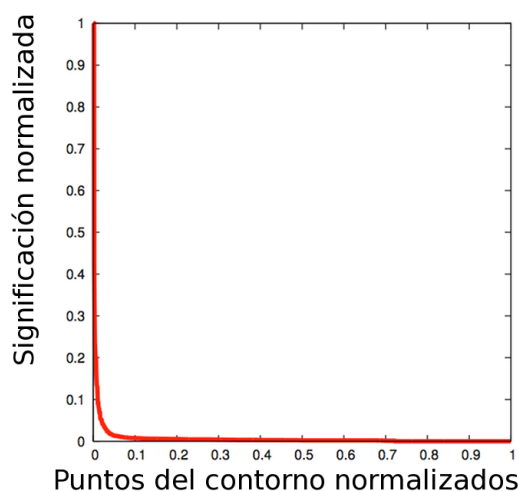


Figura 3.7: Cuarta etapa del método *split*. Generación de la curva de significación normalizada: la gráfica muestra la curva de significación normalizada generada para el contorno *chicken-5* ($N = 1364$ puntos).

3.3.5. Quinta etapa: búsqueda del umbral de los valores de significación normalizados

El objetivo de esta etapa es localizar el umbral que se corresponderá con el valor que será utilizado para umbralizar la curva de significación normalizada. Este umbral intenta contrapesar la influencia del número de puntos de la aproximación final y su error de desviación. La ordenada y del *punto de corte* P es el valor escogido h que se utilizará para umbralizar los valores de significación normalizados de los puntos del contorno (véase la sección 3.3.6).

El *punto de corte* se busca dentro de la *región de interés*, que se define por el intervalo [*Punto del extremo izquierdo*, *Punto del extremo derecho*] del eje de abscisas, donde el *Punto del extremo izquierdo* es el mayor punto normalizado con un valor de significación normalizado igual a 1, y el *Punto del extremo derecho* es el menor punto normalizado con un valor de significación normalizado igual a 0.

Se han ensayado *seis métodos de umbralización* para realizar la búsqueda del umbral de los valores de significación normalizados de los puntos del contorno. De estos seis métodos, uno consiste en una variación del método de Rosin; respecto a los otros cinco, no tenemos constancia de que se hayan utilizado en este campo de investigación con anterioridad. Cabe destacar, además, que no se tiene constancia del uso de un método de umbralización, como los que se describen a continuación, en este campo de investigación.

1. **Método de proximidad:** este método busca el punto $P(x, y)$ en la región de interés de la curva de significación normalizada cuya distancia al punto $(3/N, 0)$ sea la menor, donde N es el número de puntos del contorno. El valor 3 considera al polígono con el menor número de lados (triángulo). El umbral seleccionado se corresponde con el valor de la ordenada y del *punto de corte* o *esquina* P (véase la figura 3.8 (a)).
2. **Método de distancia:** este método busca el punto $P(x, y)$ en la región de interés de la curva de significación normalizada cuya distancia al punto $(1, 0)$ sea la máxima. El valor de la ordenada y del *punto de esquina* P se asigna como el umbral de esta etapa (véase la figura 3.8 (b)).
3. **Variante del método de Rosin:** este método se ha inspirado en el método de Rosin, que fue empleado para la umbralización unimodal de imágenes monocromáticas [91]. Este método busca el punto $P(x, y)$ en la región de interés de la curva de significación que tenga la *mayor* distancia a la recta definida por los puntos (*extremo izquierdo*, 1.0) y (*extremo derecho*, 0.0). Esta recta, denominada *recta de Rosin*, une el último punto que tiene una significación igual a 1.0 y el primer punto que tiene un valor de significación igual a 0.0. El valor de la ordenada y del *punto de esquina* P se asigna como el umbral de esta etapa (véase la figura 3.9 (a)).
4. **Método de la diagonal:** este método busca el punto $P(x, y)$ en la región de interés de la curva de significación que posea la *menor* distancia a la *diagonal* definida por los puntos $(0, 0)$ y $(1, 1)$. El punto P debe estar posicionado bajo dicha diagonal. En

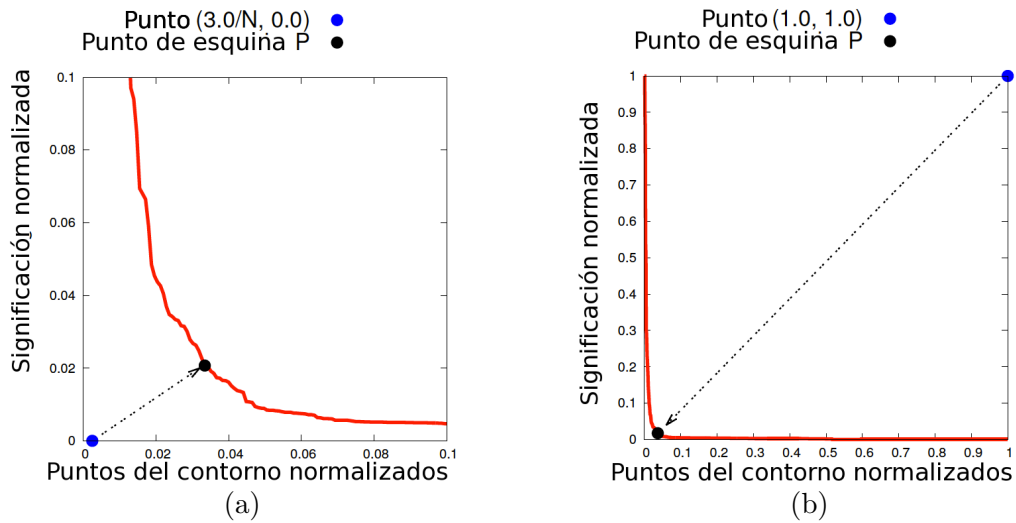


Figura 3.8: Quinta etapa del método *split*: (a) *primer método de umbralización ensayado*: el umbral es la ordenada y del punto P de la curva de significación normalizada que es más cercano al punto $(3/N, 0)$, donde N es el número de puntos del contorno. Se ha realizado un zoom en la región $[0, 0.1]$; (b) *segundo método de umbralización ensayado*: el umbral es la ordenada y del punto P de la curva de significación normalizada que está más alejado del punto $(1, 1)$.

otras palabras, se busca el primer punto $P(x, y)$ en orden ascendente que cumpla que $x > y$. El valor de la ordenada y del *punto de esquina* P se asigna como el umbral de esta etapa (véase la figura 3.9 (b)).

- Método adaptativo:** este método busca, en la región de interés de la curva de significación, el punto $P(x, y)$ más alejado del punto adaptativo $Q(x_0, 1)$, donde la proyección del punto Q , que es $Q'(x_0, 0)$, sea el primer punto, en orden ascendente, que posee un valor de significación igual a 0. El valor de la ordenada y del *punto de esquina* P se asigna como el umbral de esta etapa (véase la figura 3.10 (a)).

Se ha comprobado que, cuando el número de *break points* es bastante inferior al número de puntos del contorno, el método de umbralización *adaptativo* no funciona correctamente. En estos casos, la distancia desde el *punto adaptativo* $Q(x_0, 1)$ hasta el punto más *alejado* $P(x, y)$ es igual a 1 y mayor que la distancia desde el *punto adaptativo* hasta el *punto de corte* o *esquina* (véase la figura 3.11 (b)). El umbral h propuesto por el *método adaptativo* para estos contornos sería 0 y todos los *break points* deberían ser considerados como puntos dominantes, lo que no sería deseable. Para evitar este inconveniente, si el umbral h propuesto por el *método adaptativo* es 0 entonces se utilizará el umbral h propuesto por el *método de proximidad*.

- Método semi-adaptativo:** supone una variación del método adaptativo porque busca el punto $P(x, y)$, en la región de interés de la curva de significación, que tenga la distancia *mayor* al *punto semi-adaptativo* $Q_{half}(\frac{x_0+1}{2}, 1)$, que es el punto *intermedio* entre el *punto adaptativo* $Q(x_0, 1)$ y el punto $(1, 1)$. El valor de la ordenada y del *punto de esquina* P será el umbral de esta etapa (véase la figura 3.10 (b)).

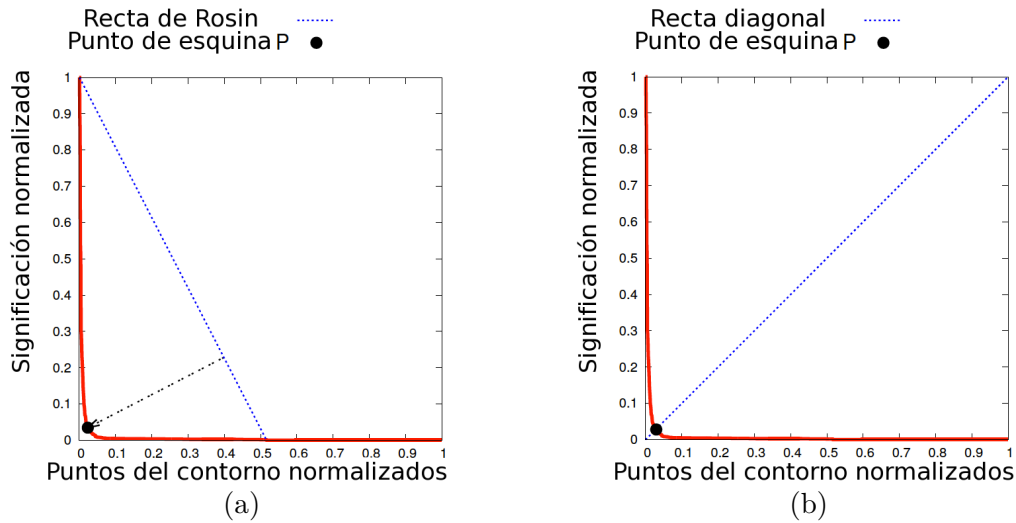


Figura 3.9: Quinta etapa del método *split*: (a) *tercer método de umbralización ensayado*: el umbral es la ordenada y del punto P más cercano a la *recta de Rosin*; (b) *cuarto método de umbralización ensayado*: el umbral es la ordenada y del punto P más cercano bajo la línea de la diagonal.

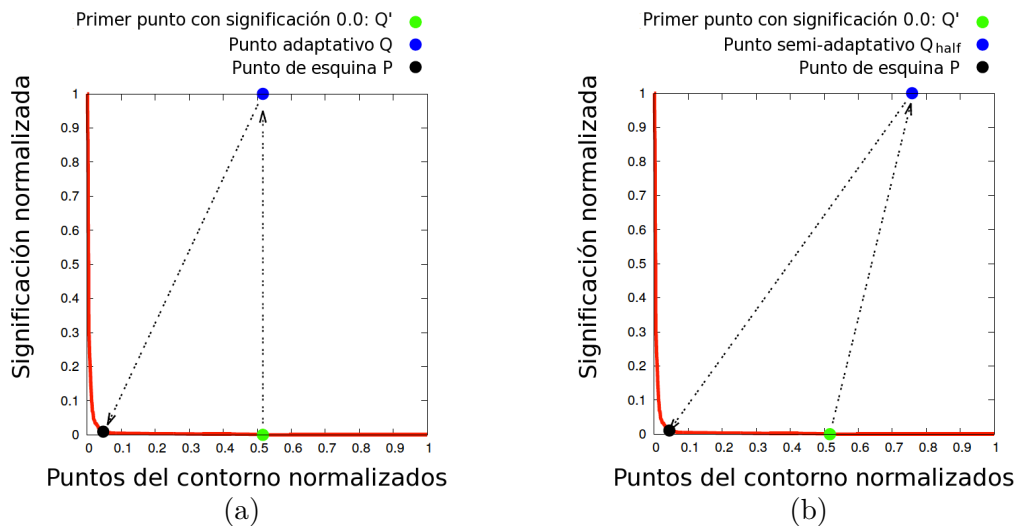


Figura 3.10: Quinta etapa del método *split*: (a) *quinto método de umbralización ensayado*: el umbral es la ordenada y del punto P más alejado del *punto adaptativo* Q ; (b) *sexto método de umbralización ensayado*: el umbral es la ordenada y del punto P más alejado del *punto semi-adaptativo* Q_{half} .

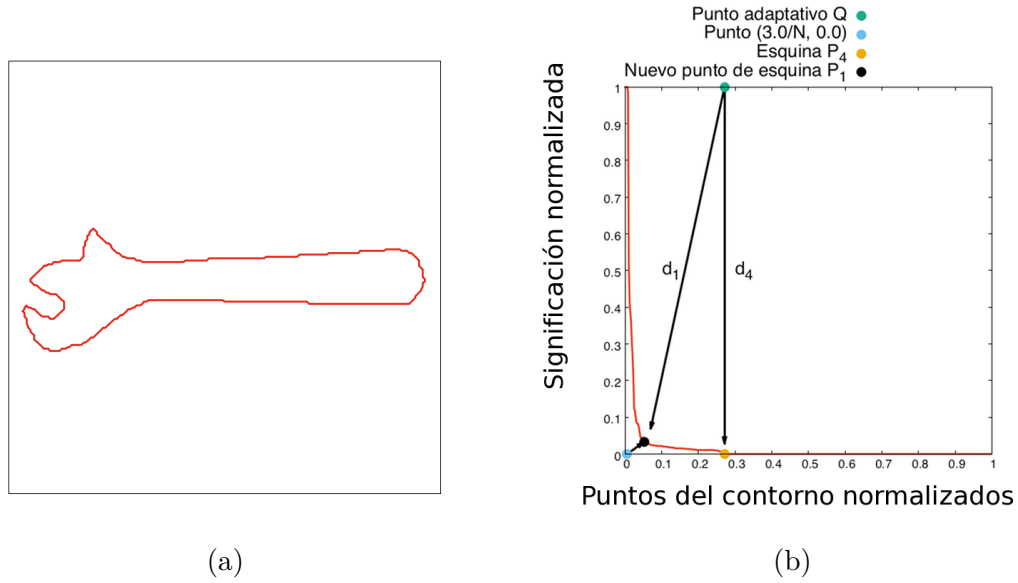


Figura 3.11: Quinta etapa del método *split*. Caso especial: (a) contorno *adjustable-spanner* ($N = 484$ puntos; solo 131 son break points = 27,07 %) y (b) curva de significación normalizada. El punto de corte P_5 propuesto por el quinto método de umbralización se cambia por el punto de corte P_1 propuesto por el primer método de umbralización. ($d_1 = 0.9919$, $d_4 = 1.0$).

En la sección 3.4.1 se evalúa el rendimiento de los métodos de umbralización ensayados. La conclusión es que el *método adaptativo*, empleando la modificación indicada para la excepción comentada en la explicación del mismo, ha obtenido los mejores resultados y, por tanto, ha sido elegido como método de umbralización de la quinta etapa de nuestra propuesta.

La complejidad computacional de esta quinta etapa es lineal $\mathcal{O}(n)$, debido a que el histograma de significación (curva de significación normalizada) se recorre de forma lineal.

3.3.6. Sexta etapa: umbralización de los valores de significación normalizados

Para obtener los *puntos dominantes* de la aproximación poligonal, se aplica un proceso de umbralización sobre los valores de significación *normalizados* de los puntos del contorno. Si $NSig(P)$ representa el valor de significación normalizado del punto P y h es el umbral propuesto por cualquiera de los métodos de umbralización propuestos (véase la sección 3.3.5), entonces los puntos dominantes del contorno serán todos aquellos puntos que tengan un valor de significación normalizado mayor que el umbral h :

$$Punto_Dominante(P) = \begin{cases} Verdadero & \text{Si } (NSig(P) > h) \text{ o } (NSig(P) = 1) \\ Falso & \text{En caso contrario} \end{cases} \quad (3.1)$$

La complejidad computacional de esta etapa es lineal $\mathcal{O}(n)$, puesto que, al igual que en la etapa anterior, el histograma de significación (curva de significación normalizada) se recorre de forma lineal.

3.4. Evaluación del método y análisis de los resultados

Se han desarrollado dos experimentos para evaluar la calidad del nuevo método de generación de aproximaciones poligonales:

- En el primer experimento (véase la sección 3.4.1) se ha evaluado el rendimiento de los seis métodos de umbralización propuestos (véase la sección 3.3.5), y se ha comprobado que el método de umbralización *adaptativo* ha obtenido los mejores resultados. Debido a ello, este método ha sido seleccionado para ser utilizado en la quinta etapa del método propuesto. Además, se ha comprobado que la excepción que se comentaba en la sección 3.3.5 no ha tenido que ser aplicada, debido a que los contornos que se han utilizado contienen un alto número de *break points*.
- En el segundo experimento (véase la sección 3.4.2) se ha comparado el nuevo método propuesto, utilizando el método de umbralización *adaptativo*, con los métodos no paramétricos propuestos por Prasad et al. [76] para detección de puntos dominantes.

Se ha utilizado la base de datos de figuras *MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)* [92] en los experimentos [42, 87]. Esta base de datos se encuentra disponible en [93] y está compuesta por 1400 figuras, clasificadas en 70 categorías, y cada categoría contiene 20 ejemplos con diferente rotación, tamaño, posición y resolución [94]. Los contornos de las figuras han sido extraídos utilizando el método implementado en la versión 2.4.6 de OpenCV. Estos contornos se ven afectados por problemas de digitalización y ruido similares a los que se presentan en situaciones reales.

Para evaluar la calidad de los diferentes métodos se han utilizado las siguientes medidas en los experimentos: número de puntos dominantes (*DP*), ratio de compresión (*CR*), error cuadrático integral (*ISE*), error máximo (E_∞) y la suma ponderada del error cuadrático ($\mu WE_2 + \sigma WE_2$) (véase la sección 2.2). También se ha utilizado la medida combinada $\mu WE_2 + \sigma WE_2$ para evaluar la estabilidad de cada uno de los métodos. Por último, se ha calculado una medida de comparación denominada *Index WE₂*, que se ha obtenido dividiendo el valor $\mu WE_2 + \sigma WE_2$ de cada versión por el valor mínimo obtenido entre todas las versiones. Este índice muestra lo mala que es una versión en comparación con la mejor (véanse las tablas 3.1 y 3.3).

3.4.1. Primer experimento: selección del mejor método de umbralización de la quinta etapa del nuevo método

El método que se propone para el cálculo de aproximaciones poligonales consta de seis etapas (véase la sección 3.3). La quinta etapa corresponde a la búsqueda del mejor umbral de los valores de significación de los puntos dominantes (véase la sección 3.3.5). Para dicha etapa, se han ensayado seis métodos de *umbralización unimodal*: *Proximidad*, *Distancia*, *Rosin*, *Diagonal*, *Adaptativo* y *Semi-adaptativo*. El objetivo de este experimento es seleccionar el mejor método de *umbralización* para la quinta etapa del nuevo método.

Teniendo en cuenta los seis métodos de umbralización propuestos, se han ejecutado seis versiones del método de generación de aproximaciones poligonales que se propone, sobre los 1400 contornos extraídos de las figuras de la base de datos de pruebas *MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)* [93]. Las figuras 3.13-3.15 muestran algunas aproximaciones poligonales que han sido generadas utilizando los seis métodos de umbralización descritos. Estas figuras muestran que el método de umbralización *adaptativo* genera la mejor aproximación poligonal posible. Para facilitar la comparaciones, se indica el número de puntos dominantes (*DP*) y el valor de la medida WE_2 de cada aproximación poligonal.

La evaluación cuantitativa de los métodos de umbralización se recoge en la tabla 3.1 y en la figura 3.12, que muestran el valor medio de las medidas de calidad, la desviación típica de la medida WE_2 , la medida combinada $\mu WE_2 + \sigma WE_2$ y el atributo *Index* WE_2 . De acuerdo con esta evaluación, el método *adaptativo* ha obtenido los mejores resultados con las medidas WE_2 y $\mu WE_2 + \sigma WE_2$ y ocupa la primera posición de la tabla, que se encuentra ordenada según el valor *Index* WE_2 . Este método de umbralización permite obtener aproximaciones poligonales con un número menor de puntos dominantes (DP) respecto a los otros métodos. Además, el ajuste al contorno mejora de forma significativa (véanse las figuras 3.13-3.15).

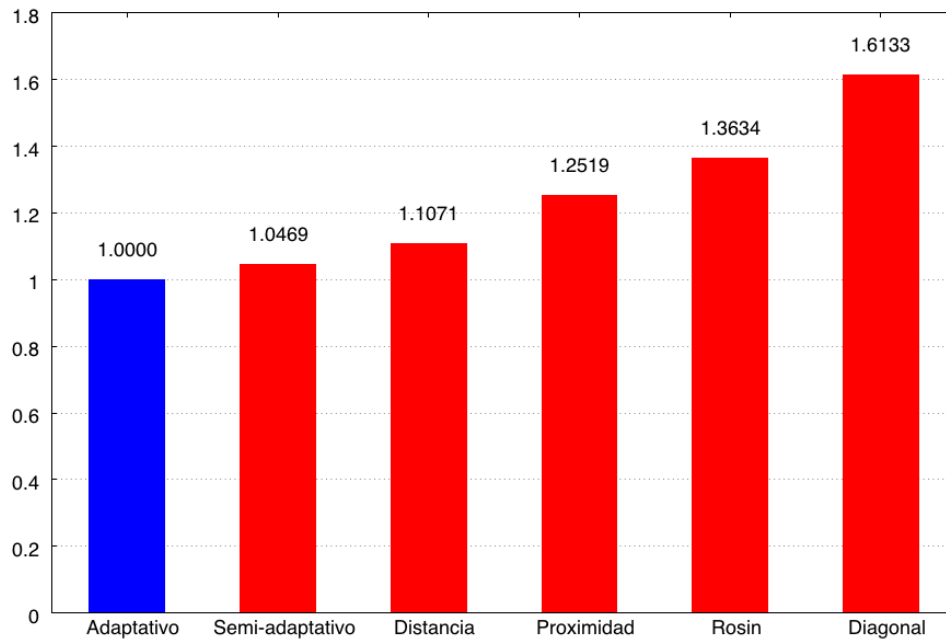
Para comprobar que existen diferencias significativas entre los métodos de umbralización, se ha aplicado el test de Wilcoxon [95] a los valores de WE_2 de las aproximaciones poligonales generadas por cada uno de dichos métodos al ser aplicados a los 1400 contornos. La tabla 3.2 muestra los resultados de las cinco pruebas que se han realizado, donde se ha comparado el método *adaptativo* con el resto de métodos de umbralización analizados. El test de Wilcoxon ha demostrado que existen diferencias significativas entre el método de umbralización *adaptativo* y el resto de los métodos propuestos.

En resumen, este primer experimento ha permitido comprobar que el método de umbralización *adaptativo* es el que ha obtenido los mejores resultados y que posee diferencias significativas con los demás métodos. Por tanto, el método de umbralización *adaptativo* será utilizado en la quinta etapa del nuevo método de generación de aproximaciones poligonales, obteniéndose un método *automático* que no requiere de ningún otro parámetro.

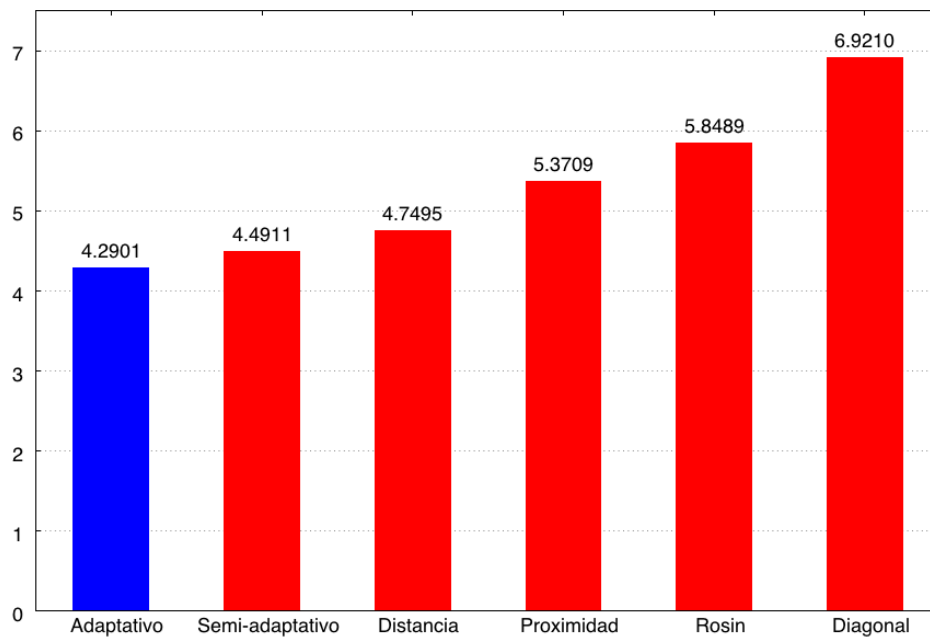
Tabla 3.1: Primer experimento. Comparativa del rendimiento del nuevo método utilizando diferentes métodos de umbralización

Orden	Versión	μ DP	μ CR	μ E $_{\infty}$	μ ISE	μ WE $_2$	σ WE $_2$	μ WE $_2 + \sigma$ WE $_2$	Index WE $_2$
1	Adaptativo	45,0693	30,9705	2,7577	2024,6730	1,7508	2,5393	4,2901	1,0000
2	Semi-adaptativo	37,5300	36,4421	3,4020	2738,7963	1,9089	2,5822	4,4911	1,0469
3	Distancia	33,1814	40,9655	3,9771	3713,3478	2,1341	2,6154	4,7495	1,1071
4	Proximidad	32,0871	41,1549	4,2525	3908,3437	2,4284	2,9425	5,3709	1,2519
5	Rosin	25,0050	56,0336	6,2181	9576,4006	2,8677	2,9812	5,8489	1,3634
6	Diagonal	31,0571	42,4564	4,8913	4331,5297	2,9193	4,0017	6,9210	1,6133

Nota: el promedio del número de puntos de los contornos de la base de datos MPEG-7 CE-Shape-1 (Part B) es 1271,04 puntos.

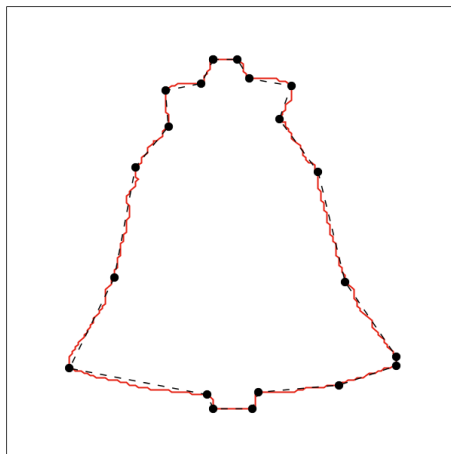


(a) Rendimiento de los métodos de umbralización. Medida $Index WE_2$

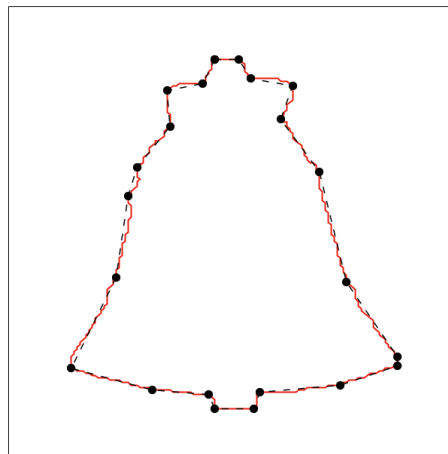


(b) Rendimiento de los métodos de umbralización. Medida $\mu WE_2 + \sigma WE_2$

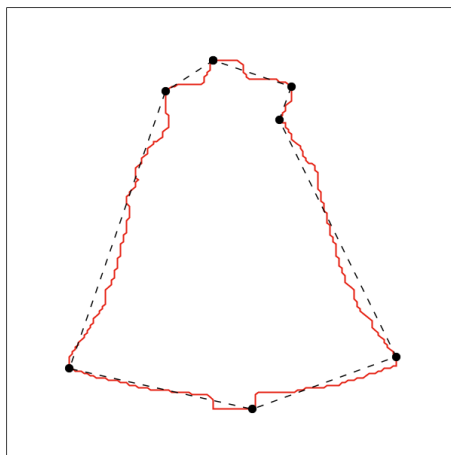
Figura 3.12: Primer experimento. Gráficas comparativas del rendimiento del nuevo método utilizando diferentes métodos de umbralización



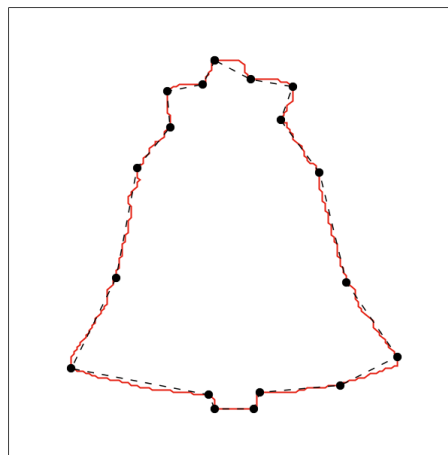
(a) *Proximidad*
 DP = 20, $WE_2 = 0,6193$



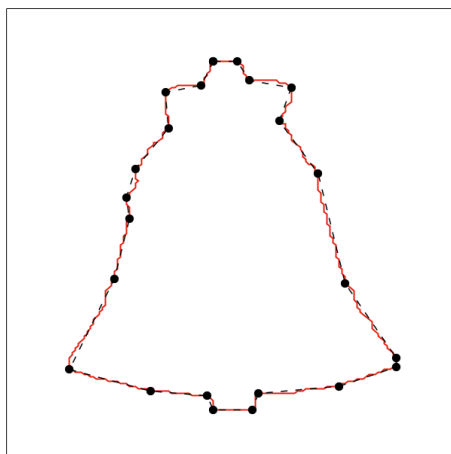
(b) *Distancia*
 DP = 22, $WE_2 = 0,5908$



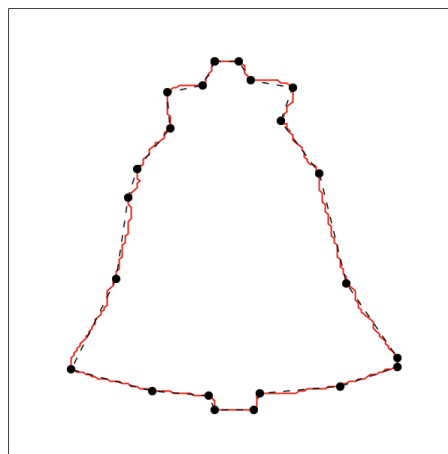
(c) *Rosin*
 DP = 7, $WE_2 = 0,6594$



(d) *Diagonal*
 DP = 18, $WE_2 = 0,8659$

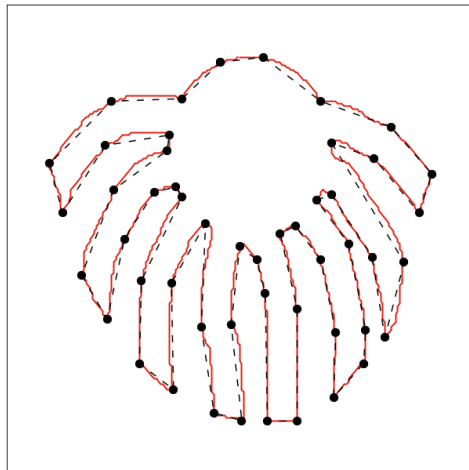


(e) *Adaptativo*
 DP = 23, $WE_2 = 0,5334$

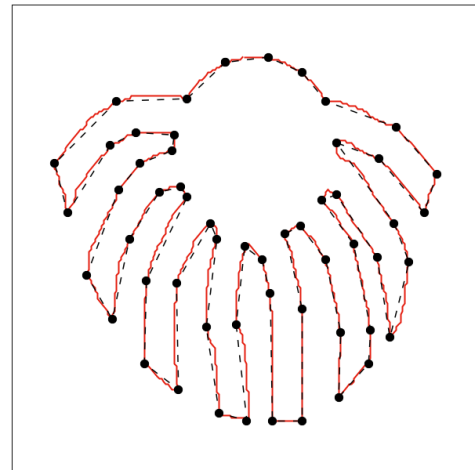


(f) *Semi-adaptativo*
 DP = 22, $WE_2 = 0,5908$

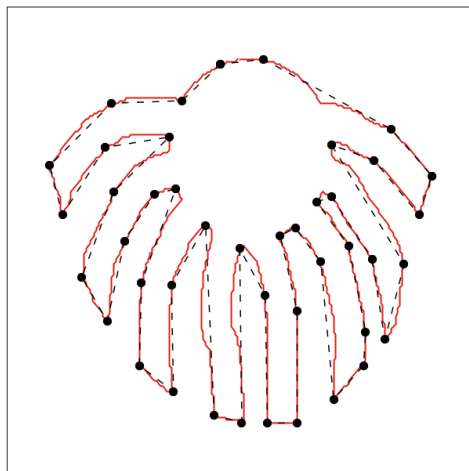
Figura 3.13: Primer experimento. Selección del mejor método de umbralización de la quinta etapa del nuevo método. Aproximaciones poligonales del contorno *bell-7* ($N = 406$ puntos).



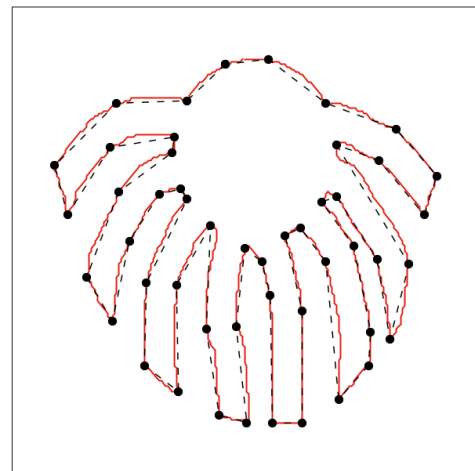
(a) *Proximidad*
 $DP = 50, WE_2 = 3,1546$



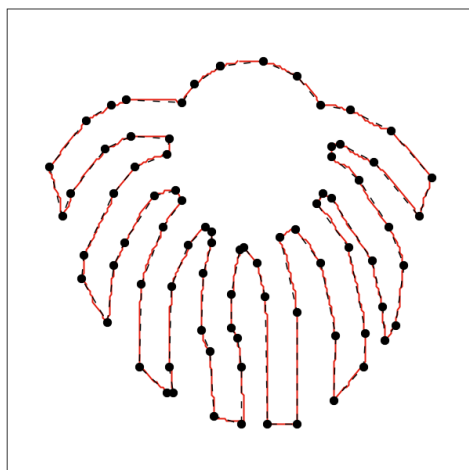
(b) *Distancia*
 $DP = 55, WE_2 = 2,6223$



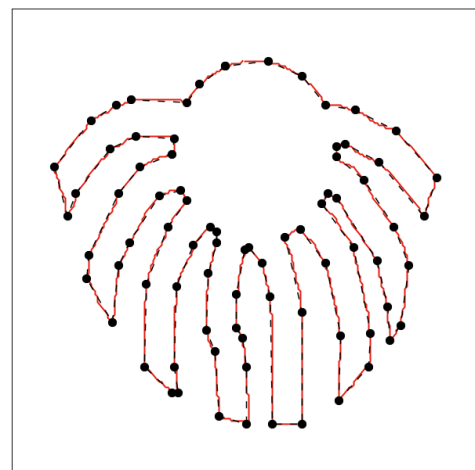
(c) *Rosin*
 $DP = 43, WE_2 = 3,3029$



(d) *Diagonal*
 $DP = 49, WE_2 = 3,4627$

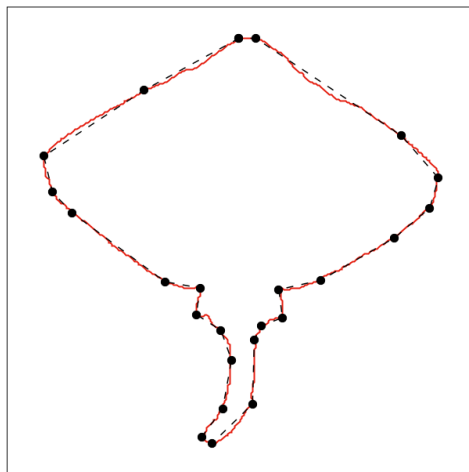


(e) *Adaptativo*
 $DP = 79, WE_2 = 1,0304$

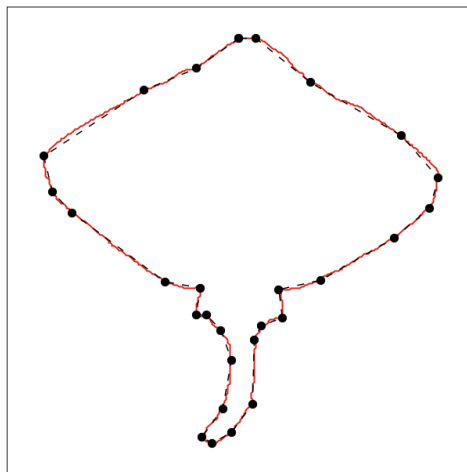


(f) *Semi-adaptativo*
 $DP = 78, WE_2 = 1,0309$

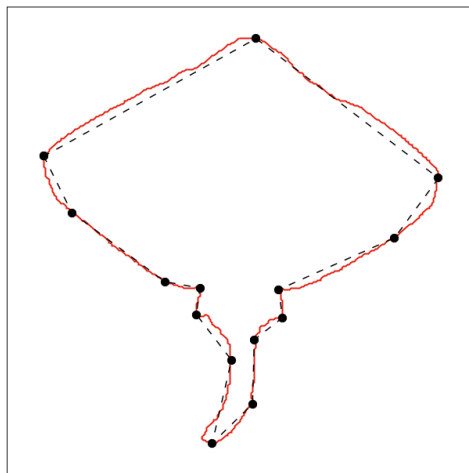
Figura 3.14: Primer experimento. Selección del mejor método de umbralización de la quinta etapa del nuevo método. Aproximaciones poligonales del contorno *octopus-14* ($N = 1211$ puntos).



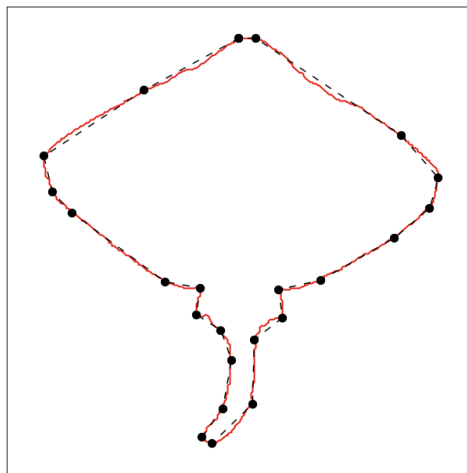
(a) *Proximidad*
 $DP = 24, WE_2 = 1,4050$



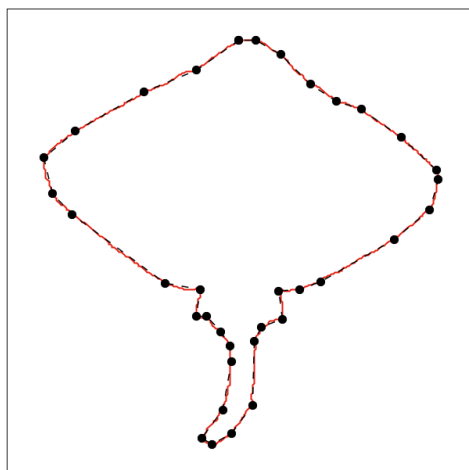
(b) *Distancia*
 $DP = 28, WE_2 = 1,0998$



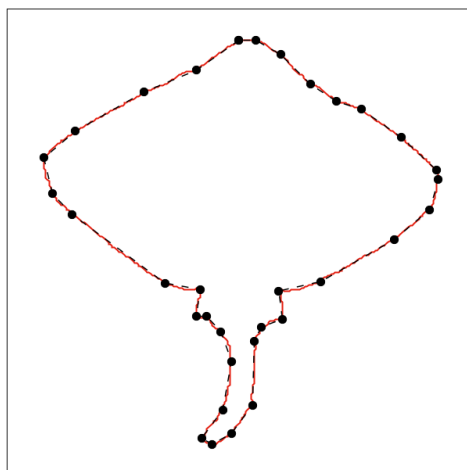
(c) *Rosin*
 $DP = 14, WE_2 = 2,8965$



(d) *Diagonal*
 $DP = 23, WE_2 = 1,4490$



(e) *Adaptativo*
 $DP = 35, WE_2 = 0,6256$



(f) *Semi-adaptativo*
 $DP = 33, WE_2 = 0,6421$

Figura 3.15: Primer experimento. Selección del mejor método de umbralización de la quinta etapa del nuevo método. Aproximaciones poligonales del contorno *ray-17* ($N = 689$ puntos).

Tabla 3.2: Test de Wilcoxon aplicado a los valores de WE_2 de las 1400 aproximaciones poligonales generadas por cada uno de los métodos de umbralización evaluados en el primer experimento. Este test pone de manifiesto la existencia de diferencias significativas entre el método número 5 (método *adaptativo*) y el resto de los métodos propuestos. En todos los casos, el nivel de significación es igual a 0.0.

Comparación WE_2	Dif. positivas	Dif. negativas	Empates
Método 1 - Método 5	1041 (74,36 %)	216 (15,43 %)	143 (10,21 %)
Método 2 - Método 5	926 (66,14 %)	298 (21,29 %)	176 (12,57 %)
Método 3 - Método 5	1189 (84,93 %)	171 (12,21 %)	40 (2,86 %)
Método 4 - Método 5	1236 (88,29 %)	134 (9,57 %)	30 (2,14 %)
Método 6 - Método 5	712 (50,86 %)	338 (24,14 %)	350 (25 %)

3.4.2. Segundo experimento: comparación del método propuesto con los métodos no-paramétricos propuestos por Prasad

En el segundo experimento se ha comparado el método propuesto (utilizando el método *adaptativo* como mecanismo de umbralización) con las versiones automáticas de los métodos *RDP*, *Carmona* y *Masood* propuestos por Prasad et al. [76]. Se han seleccionado estos métodos porque pueden generar aproximaciones poligonales de contornos planos y cerrados de forma *automática*; es decir, no paramétrica.

Estos cuatro métodos se han aplicado a los 1400 contornos extraídos de la base de datos *MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)* [92] para analizar sus rendimientos. Las figuras 3.17-3.21 muestran algunas aproximaciones poligonales generadas por estos métodos. Además, se ha empleado el conocido contorno *semicircles* [81] para mostrar el comportamiento de los métodos con contornos simétricos (véase la figura 3.22). Para facilitar la comparación del rendimiento en cada contorno, las figuras indican el número de puntos dominantes (*DP*) y el valor de la medida WE_2 .

La evaluación cuantitativa de los cuatro métodos se recoge en la tabla 3.3 y en la figura 3.16, que muestran el valor medio de las medidas de calidad, la desviación típica de la medida WE_2 , la medida combinada $\mu WE_2 + \sigma WE_2$ y el atributo *Index WE_2* . De acuerdo con esta evaluación, el nuevo método propuesto, que incluye el método de umbralización *adaptativo*, ha obtenido los mejores resultados con las medidas WE_2 y $\mu WE_2 + \sigma WE_2$ y ocupa la primera posición de la tabla, que se encuentra ordenada según el valor *Index WE_2* . Esta tabla también muestra que el nuevo método genera muchos menos puntos dominantes (*DP*) que el resto de los métodos analizados.

Las figuras 3.17 (d) - 3.21 (d) muestran que las aproximaciones poligonales que

se han obtenido con el método propuesto se ajustan adecuadamente a los contornos. Sin embargo, el rendimiento de los métodos propuestos por Prasad et. al. [76] depende del contorno. Por ejemplo, estos métodos generan demasiados puntos dominantes para algunos contornos (véanse las figuras 3.17, 3.19 y 3.20) y muy pocos puntos dominantes para otros contornos (véanse las figuras 3.20 (c) y 3.21 (b)-(c)).

La nueva propuesta no se ve afectada por el ruido de digitalización de los contornos que han sido utilizados en los experimentos (véase, por ejemplo, la figura 3.18). Además, el nuevo método sí tiene en cuenta la simetría de los contornos, mientras que las versiones propuestas por Prasad et. al. no tienen esta propiedad. Véanse las aproximaciones poligonales generadas para el contorno *semicircles* en la figura 3.22.

En nuestra opinión, este experimento demuestra, una vez más, que la medida WE (la inversa de la medida FOM) no es la mejor medida para comparar aproximaciones poligonales con diferente número de puntos dominantes, puesto que favorece las aproximaciones poligonales con un menor ISE , que pueden obtenerse fácilmente incrementando el número de puntos dominantes [19].

3.5. Conclusiones

En este capítulo se ha propuesto un nuevo método *automático* para generar aproximaciones poligonales de contornos bidimensionales cerrados. El método propuesto computa los valores de significación de los puntos utilizando una nueva versión del método RDP [52, 84] que tiene en cuenta la *simetría* del contorno. Además, el método se basa en una nueva estrategia que aplica un método *adaptativo* para umbralizar los valores de significación normalizados de los puntos del contorno para obtener los puntos dominantes de la aproximación poligonal.

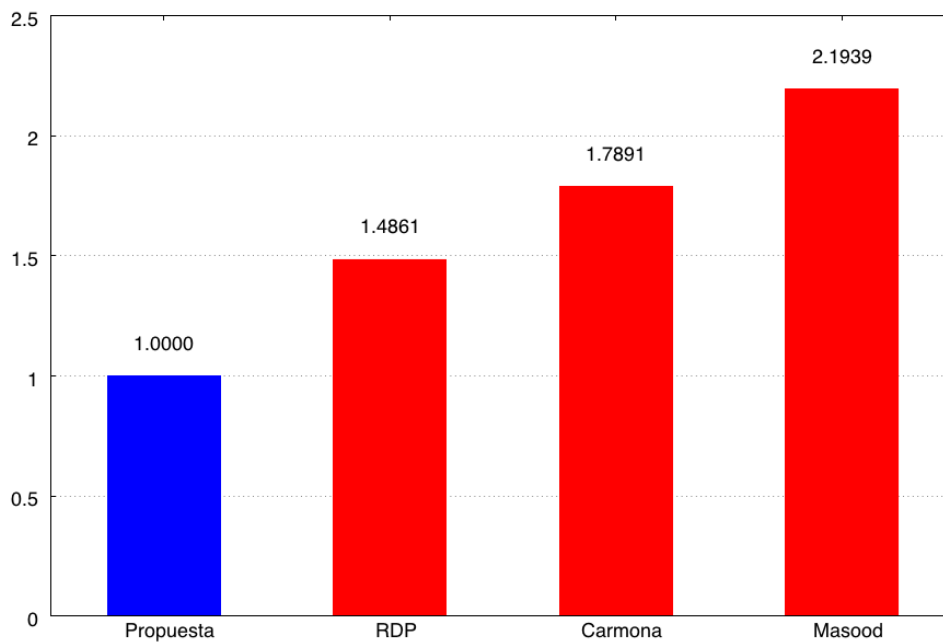
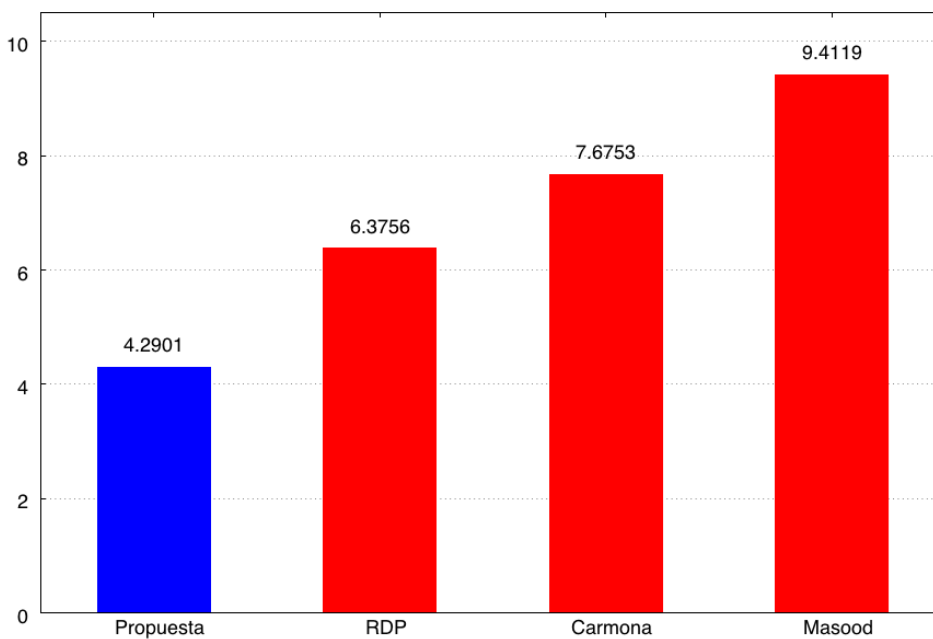
El nuevo método tiene una complejidad computacional de $\mathcal{O}(n \log(n))$ en el mejor de los casos y de $\mathcal{O}(n^2)$ en el peor, si todos los *break points* son puntos dominantes. Los experimentos han probado que el nuevo método puede generar aproximaciones poligonales eficientes y efectivas para contornos cerrados de objetos bidimensionales.

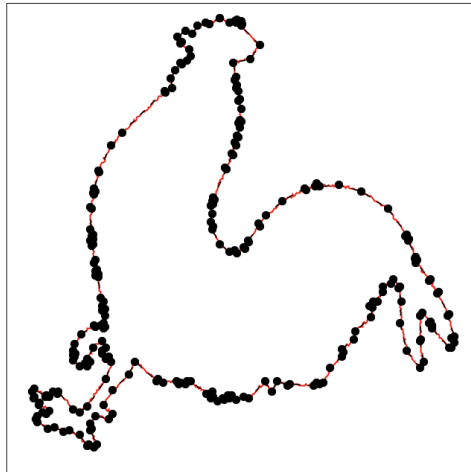
El método propuesto selecciona los puntos dominantes de la aproximación poligonal umbralizando sus niveles de significación. Debido a ello, puede ser aplicado con cualquier algoritmo que asigne un valor de significación a los puntos del contorno. Este método ha sido publicado en un artículo de la revista *Journal of Visual Communication and Image Representation* en el año 2016 [96] y también puede ser consultado en el anexo A.

Tabla 3.3: Segundo experimento. Comparativa del rendimiento del nuevo método utilizando diferentes métodos de umbralización

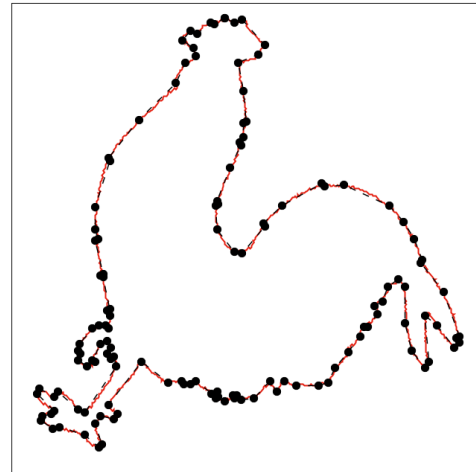
Orden	Versión	μ DP	μ CR	μ E $_{\infty}$	μ ISE	μ WE $_2$	σ WE $_2$	μ WE $_2 + \sigma$ WE $_2$	Index WE $_2$
1	Propuesta	45,0693	30,9705	2,7577	2024,6730	1,7508	2,5393	4,2901	1,0000
2	RDP	110,1185	17,7756	1,2271	219,3423	2,2658	4,1098	6,3756	1,4861
3	Carmona	67,6978	32,5860	3,6299	2168,1644	3,1639	4,5114	7,6753	1,7891
4	Masood	131,9271	39,5855	5,0971	92491,0892	3,4491	5,9628	9,4119	2,1939

Nota: el promedio del número de puntos de los contornos de la base de datos MPEG-7 CE-Shape-1 (Part B) es 1271,04 puntos.

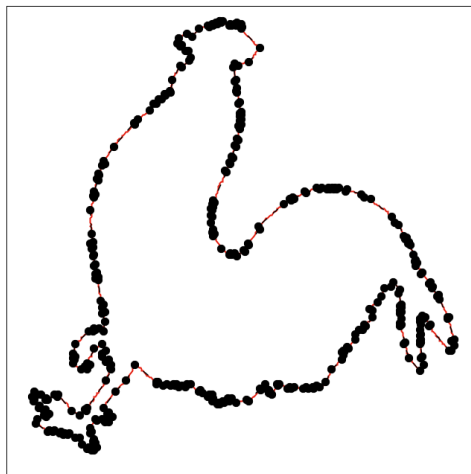
(a) Rendimiento del método propuesto. Medida $Index WE_2$ (b) Rendimiento del método propuesto. Medida $\mu WE_2 + \sigma WE_2$ **Figura 3.16:** Segundo experimento. Tabla comparativa del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto.



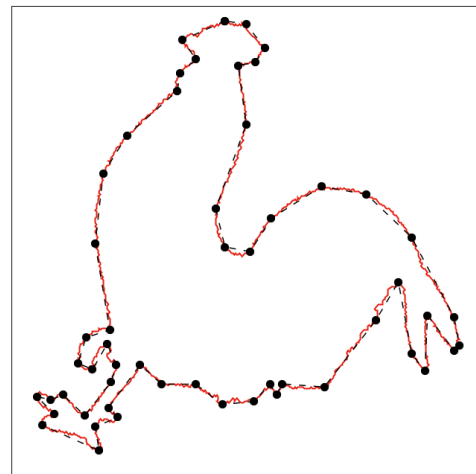
(a) RDP
 $DP = 255, WE_2 = 9,9899$



(b) Carmona
 $DP = 134, WE_2 = 8,7495$

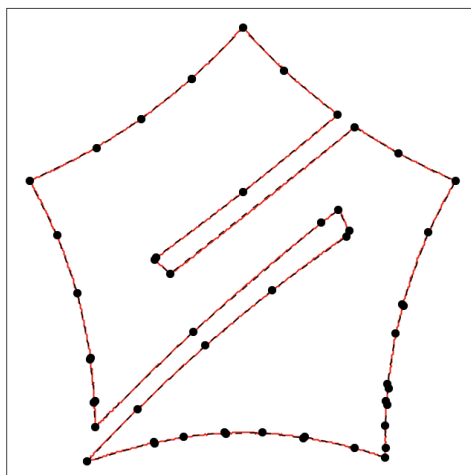


(c) Masood
 $DP = 401, WE_2 = 12,7870$

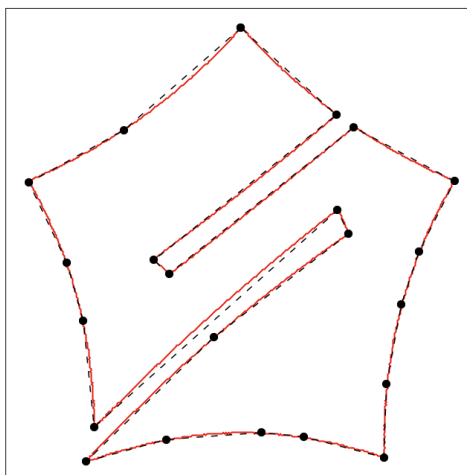


(d) *Método propuesto*
 $DP = 54, WE_2 = 3,8040$

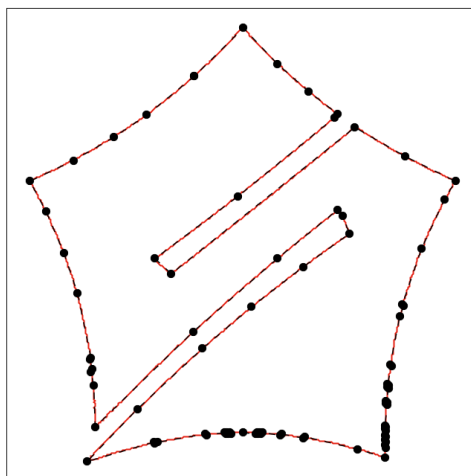
Figura 3.17: Segundo experimento. Comparación del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto. Aproximaciones poligonales del contorno *chicken-5* ($N = 1364$ puntos).



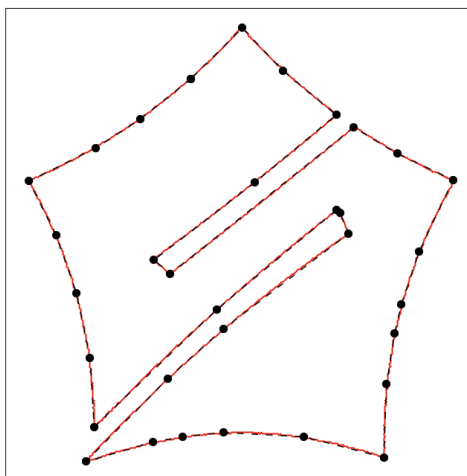
(a) RDP
 $DP = 50, WE_2 = 0,3052$



(b) Carmona
 $DP = 22, WE_2 = 0,6544$

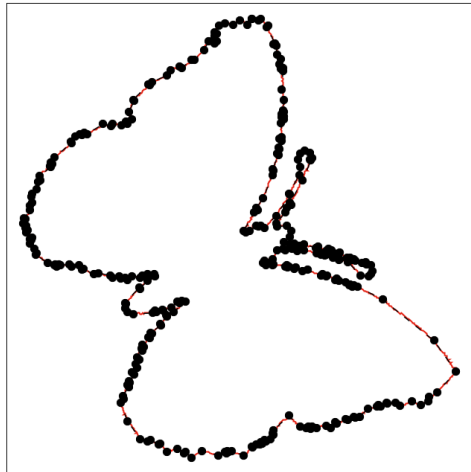


(c) Masood
 $DP = 84, WE_2 = 0,5312$

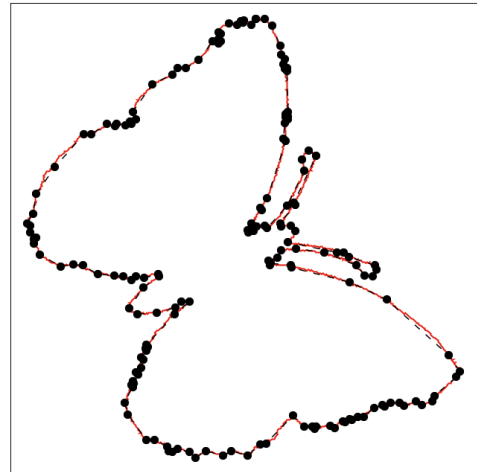


(d) *Método propuesto*
 $DP = 33, WE_2 = 0,1595$

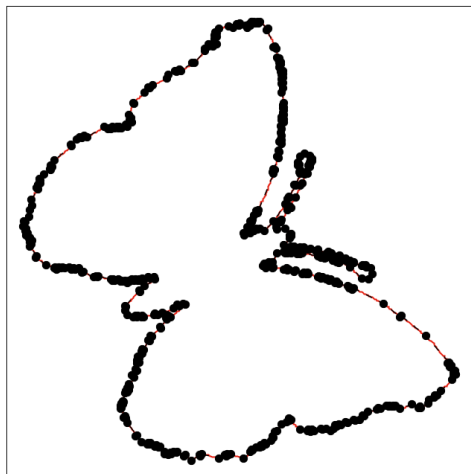
Figura 3.18: Segundo experimento. Comparación del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto. Aproximaciones poligonales del contorno *device6-9* ($N = 1590$ puntos).



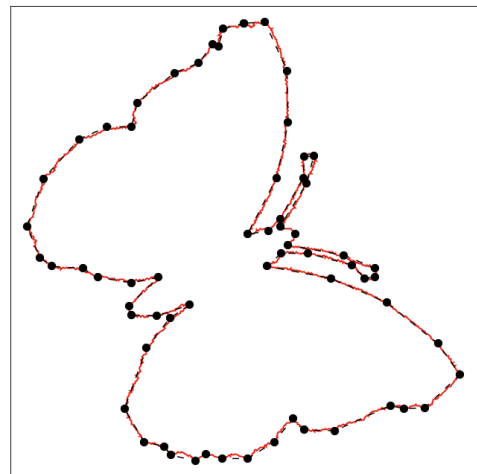
(a) RDP
 $DP = 339, WE_2 = 13,9270$



(b) Carmona
 $DP = 171, WE_2 = 13,3103$

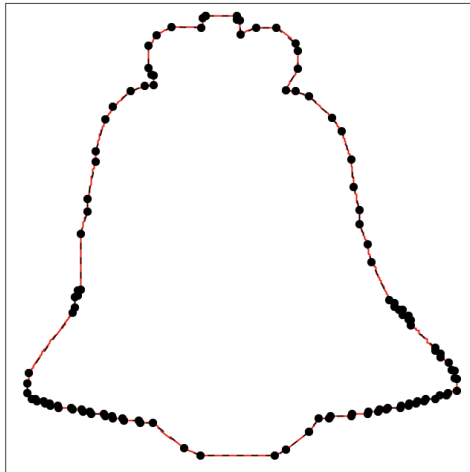


(c) Masood
 $DP = 525, WE_2 = 17,2215$

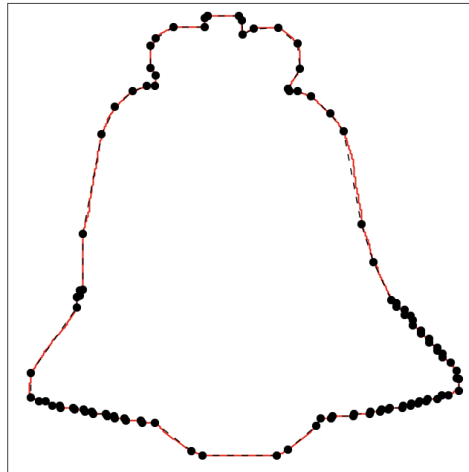


(d) *Método propuesto*
 $DP = 65, WE_2 = 2,9114$

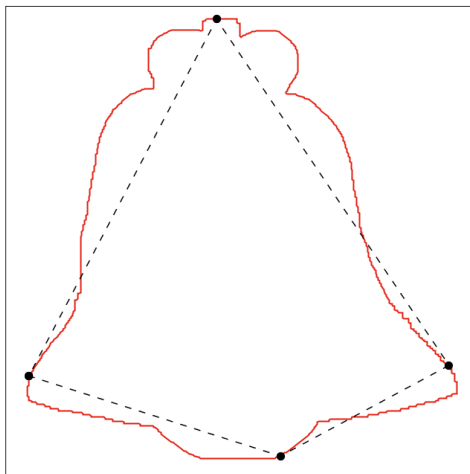
Figura 3.19: Segundo experimento. Comparación del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto. Aproximaciones poligonales del contorno *butterfly-13* ($N = 1786$ puntos).



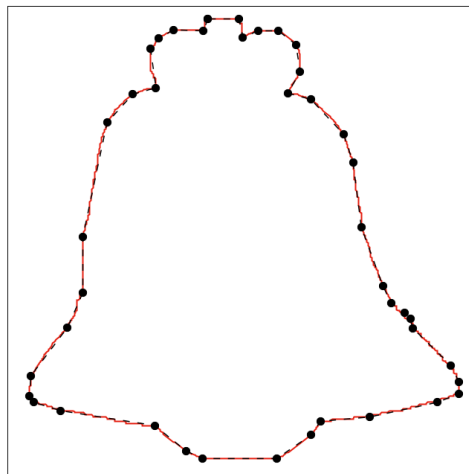
(a) RDP
 $DP = 120, WE_2 = 1,5530$



(b) Carmona
 $DP = 104, WE_2 = 3,9602$

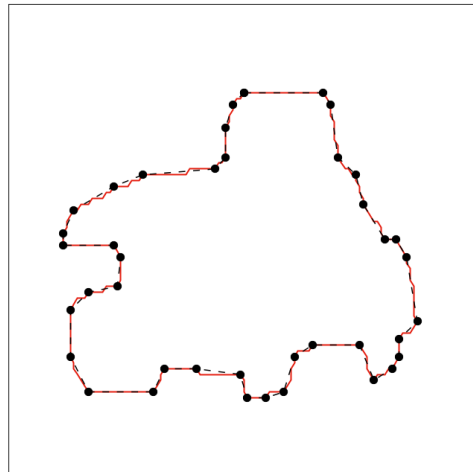


(c) Masood
 $DP = 4, WE_2 = 4,9548$

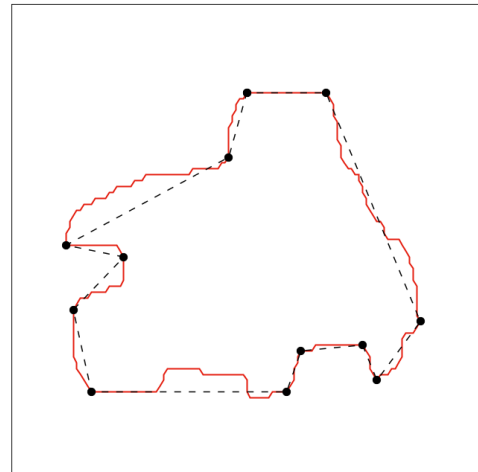


(d) *Método propuesto*
 $DP = 42, WE_2 = 0,8483$

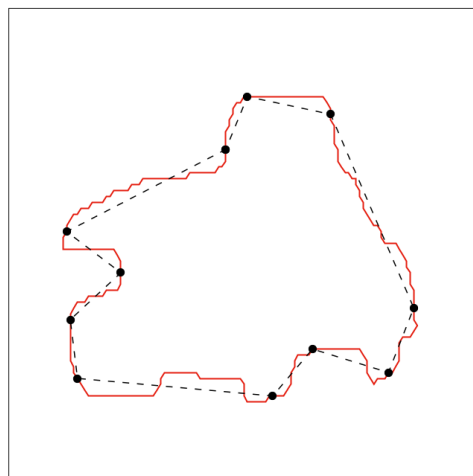
Figura 3.20: Segundo experimento. Comparación del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto. Aproximaciones poligonales del contorno *bell-10* ($N = 1202$ puntos).



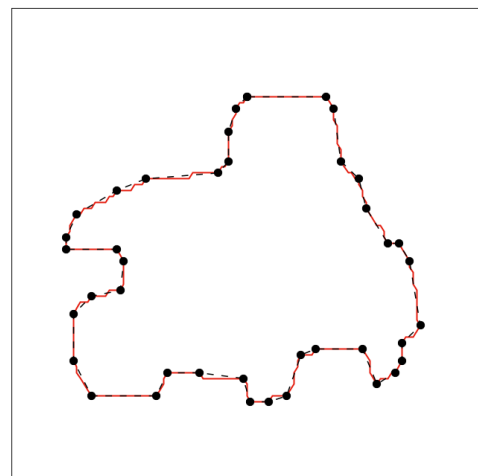
(a) RDP
 $DP = 40, WE_2 = 0,5063$



(b) Carmona
 $DP = 12, WE_2 = 2,1101$

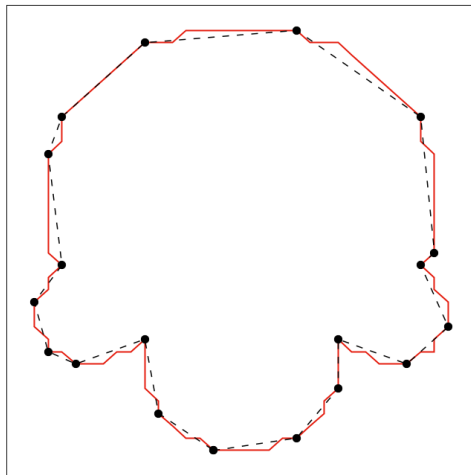


(c) Masood
 $DP = 11, WE_2 = 1,293$

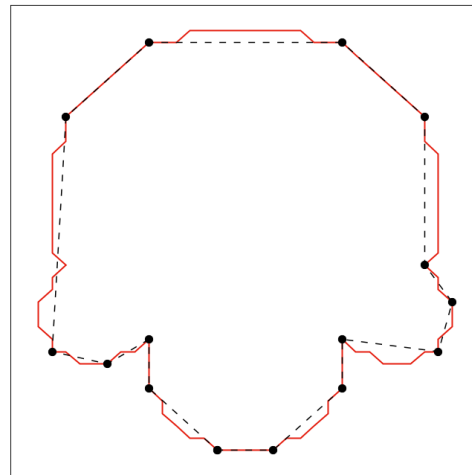


(d) *Método propuesto*
 $DP = 40, WE_2 = 0,5001$

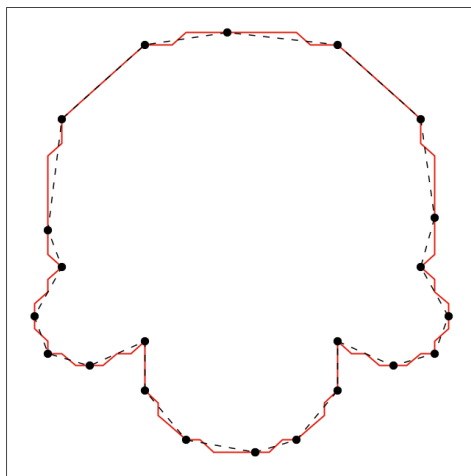
Figura 3.21: Segundo experimento. Comparación del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto. Aproximaciones poligonales del contorno *truck-07* ($N = 277$ puntos).



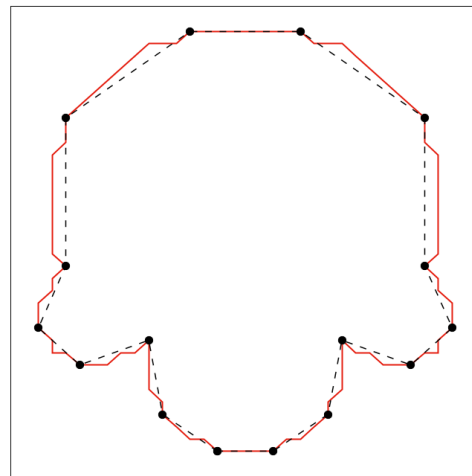
(a) RDP
 $DP = 19, WE_2 = 0,5518$



(b) Carmona
 $DP = 15, WE_2 = 0,8140$



(c) Masood
 $DP = 21, WE_2 = 0,3941$



(d) Método propuesto
 $DP = 16, WE_2 = 0,7711$

Figura 3.22: Segundo experimento. Comparación del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto. Aproximaciones poligonales del contorno *semicircles* ($N = 102$ puntos). Solo el método *propuesto* genera una aproximación poligonal simétrica del contorno *semicircles*

4

Análisis exhaustivo del nuevo método de obtención de aproximaciones poligonales

4.1. Introducción

El capítulo anterior ha presentado y descrito un nuevo método para la obtención de aproximaciones poligonales que está basado en la estrategia de división (*split*) del contorno en sus partes más representativas mediante la elección de los puntos dominantes más significativos. El método propuesto consta de varias fases y algunas de ellas permiten introducir alternativas para crear nuevas versiones. En este capítulo se describen y evalúan las alternativas que pueden ser empleadas para realizar un estudio exhaustivo del método explicado en el capítulo anterior. Las alternativas analizadas se pueden clasificar en dos grandes grupos o categorías:

- Basadas en el criterio de *división* de puntos del contorno: estrategia *split*.
- Basadas en el criterio de *fusión* de puntos del contorno: estrategia *merge*.

En la sección 2.1, del capítulo 2 de antecedentes, se pueden consultar los fundamentos teóricos básicos para comprender las estrategias *split* y *merge* de generación de aproximaciones poligonales.

La organización capítulo es la siguiente: en la sección 4.2 se explican las alternativas del método propuesto que están basadas en la estrategia *split* y la sección 4.3 desarrolla las modificaciones del método basadas en la estrategia *merge*; los experimentos realizados y los resultados obtenidos son descritos en la sección 4.4; por último, la sección 4.5 resume las principales conclusiones.

4.2. Alternativas basadas en la estrategia *split*

El método de generación de aproximaciones poligonales propuesto en el capítulo 3 está basado en una estrategia *split*. Describimos a continuación cada una de las alternativas consideradas, de las que puede verse un resumen en la figura 4.1:

1. *Selección de puntos iniciales*: en esta etapa del método se han considerado cuatro alternativas diferentes (véase la sección 4.2.1).
2. *Computación de los valores de significación de los puntos que no son iniciales*: se han propuesto tres medidas para calcular la significación de los puntos no iniciales del contorno (véase la sección 4.2.2).
3. *Computación de los valores de significación de los puntos iniciales*: se contemplan dos alternativas posibles a la hora de calcular dicha significación (véase la sección 4.2.3).
4. *Generación de la curva de significación normalizada*: el proceso para obtener la curva de significación normalizada no ha sido modificado en este estudio, sino que se mantiene la propuesta original (véase la sección 3.3.4).
5. *Búsqueda del umbral de los valores de significación normalizados*: se utilizarán los métodos de umbralización propuestos en la sección 3.3.5 del capítulo 3.
6. *Umbralización de la curva de significación normalizada*: el proceso utilizado para umbralizar los valores de significación es idéntico al que se adoptó en el nuevo método *split* del capítulo 3, descrito en la sección 3.3.6.
7. *Borrado de puntos dominantes superfluos generados por el casco convexo (opcional)*: esta última etapa solo se aplica si se ha utilizado la alternativa del *casco convexo* para seleccionar los puntos iniciales en la primera etapa del método. En algunos casos, la versión basada en el *casco convexo* puede generar una aproximación poligonal que contiene demasiados puntos dominantes *superfluos* en aquellas partes del contorno que tienen una forma *redondeada*. Esta fase de borrado suprimirá todos aquellos puntos dominantes que estén *casi alineados* (véase la sección 4.2.4).

En las secciones 4.2.1-4.2.4 se detallan los cambios introducidos en las etapas primera, segunda, tercera y séptima de este método respecto al método original.

4.2.1. Primera etapa: selección de puntos iniciales

En la primera etapa del método se lleva a cabo la selección de los *puntos iniciales* del contorno. Se han considerado cuatro alternativas:

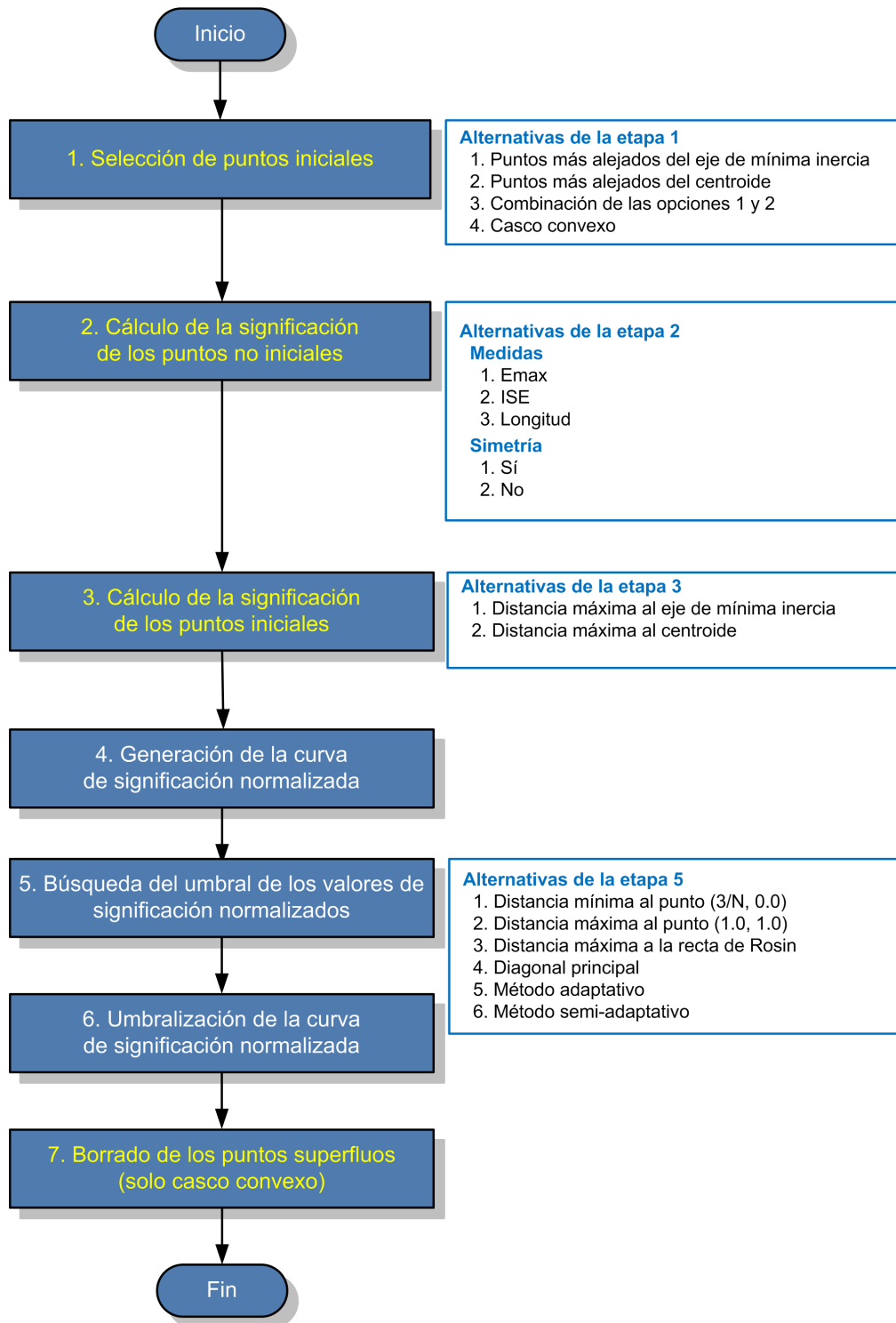


Figura 4.1: Esquema del método de la estrategia *split*. Se resaltan en amarillo las etapas que se han modificado o añadido al método original.

1. Seleccionar los puntos más alejados del *eje de mínima inercia* y los puntos más alejados de los anteriores.
2. Seleccionar los puntos más alejados del *centroide* y los puntos más alejados de los anteriores.
3. *Combinación* de las alternativas 1 y 2.
4. Puntos del *casco convexo* del contorno.

Todas las alternativas propuestas en esta etapa son invariantes a rotaciones y cambios de escala y no dependen de la elección del primer punto en la secuenciación o parametrización de los puntos del contorno. Además, también tienen en cuenta la simetría del contorno. La complejidad computacional de los dos primeros métodos es lineal $\mathcal{O}(n)$, mientras que la complejidad computacional del método basado en el *casco convexo* es $\mathcal{O}(n \log(n))$. Si los puntos se encuentran ordenados por una de las coordenadas o por el ángulo a un vector fijo, entonces la complejidad del método basado en el *casco convexo* es $\mathcal{O}(n)$ [97, 98].

Alternativa 1. Puntos más alejados del eje de mínima inercia

Esta alternativa selecciona como *puntos iniciales* a los puntos del contorno que estén más alejados del eje de mínima inercia. Si la distancia máxima al eje de mínima inercia se alcanza en dos o más puntos, entonces solamente serán considerados como *puntos iniciales* los que de ellos tengan la máxima distancia al centroide. Por último, también serán considerados como *puntos iniciales* los que tengan la máxima distancia a los *puntos iniciales* obtenidos previamente.

Esta alternativa consiste en los siguientes pasos (véase la figura 4.2 (a)):

1. Calcular el *centroide*¹ y el *eje de mínima inercia*² del contorno.
2. Sea d_{ext} la *distancia máxima al eje de mínima inercia*.

Inicialmente, $d_{max} \leftarrow 0.0$.

3. Sea dc_{max} la *distancia máxima al centroide* entre los puntos que tienen la *distancia máxima al eje de mínima inercia*.

Inicialmente, $dc_{max} \leftarrow 0.0$.

4. Para cada punto del contorno P_i ($i \in \{1, \dots, n\}$):

¹El *centro de gravedad* o *centroide* de un contorno determinado por el conjunto de puntos $(x_i, y_i) \forall i \in \{1, \dots, n\}$ es el punto (\bar{x}, \bar{y}) definido por las siguientes expresiones:

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N} \quad ; \quad \bar{y} = \frac{\sum_{i=1}^N y_i}{N}$$

²El *eje de mínima inercia* de un conjunto de puntos es una línea recta que: (1) pasa por el *centroide* de los puntos y (2) la suma de las distancias al cuadrado desde cada punto a dicha recta es *mínima*.

Calcular la distancia d_i desde el punto P_i hasta el *eje de mínima inercia*:

a) Si $(d_i > d_{max})$ entonces:

$$d_{max} \leftarrow d_i \text{ y } dc_{max} \leftarrow d(P_i, C),$$

donde $d(P_i, C)$ es la distancia del punto P_i al centroide C del contorno.

b) Si $(d_i = d_{max})$ entonces:

$$\text{si } (dc_{max} > d(P_i, C)) \text{ entonces } dc_{max} \leftarrow d(P_i, C)$$

(Se actualiza la máxima distancia al centroide entre los puntos que tienen la máxima distancia al eje de mínima inercia.)

5. Para cada punto del contorno P_i ($i \in \{1, \dots, n\}$):

si $(d_i = d_{max})$ y $(d(P_i, C) = dc_{max})$ entonces

marcar al punto P_i como *punto inicial*.

6. Finalmente, considerar también como *puntos iniciales* a los puntos más alejados de los *puntos iniciales* obtenidos en el paso anterior.

Alternativa 2. Puntos más alejados del centroide

Este método selecciona como *puntos iniciales* a los puntos más alejados del centroide y, además, considera también como *puntos iniciales* a los puntos más alejados de los puntos anteriores.

Esta es la alternativa que ya fue propuesta en el nuevo método descrito en el capítulo 3 (véase la sección 3.3.1 y las figuras 3.3 y 4.2 (b)).

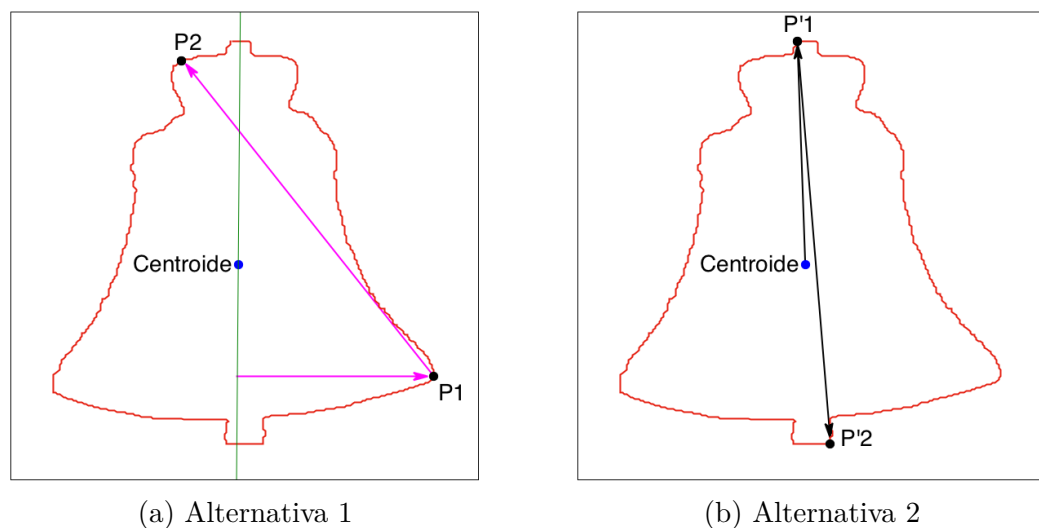


Figura 4.2: Primera etapa: selección de *puntos iniciales*. (a) Alternativa 1: el punto P_1 es el más alejado del *eje de mínima inercia* y P_2 es el punto más alejado del punto P_1 ; (b) Alternativa 2: P'_1 es el punto más alejado del centroide del contorno y P'_2 es el punto más alejado del punto P'_1 .

Alternativa 3. Combinación de los métodos 1 y 2

Esta alternativa utiliza los dos métodos propuestos previamente, para considerar como *puntos iniciales* a los puntos más alejados del *eje de mínima inercia* y a los puntos más alejados del *centroide*. Además, también considera como *puntos iniciales* a los puntos más alejados de los puntos anteriores (véase la figura 4.3 (a)).

Alternativa 4. Puntos del casco convexo

La cuarta alternativa calcula el *casco convexo* del contorno [97, 98]. El *casco convexo* es el polígono con menor área que envuelve al contorno. Los vértices del *casco convexo* serán considerados como *puntos iniciales* (véase la figura 4.3 (b)).

Si el contorno tiene partes redondeadas, este método puede proporcionar demasiados puntos iniciales, lo que generará una gran cantidad de puntos dominantes en la aproximación final (véase la figura 4.4). Debido a esto, se aplicará la etapa adicional de *borrado*, que *suprimirá los puntos dominantes superfluos introducidos por la aplicación del casco convexo* (véase la sección 4.2.4).

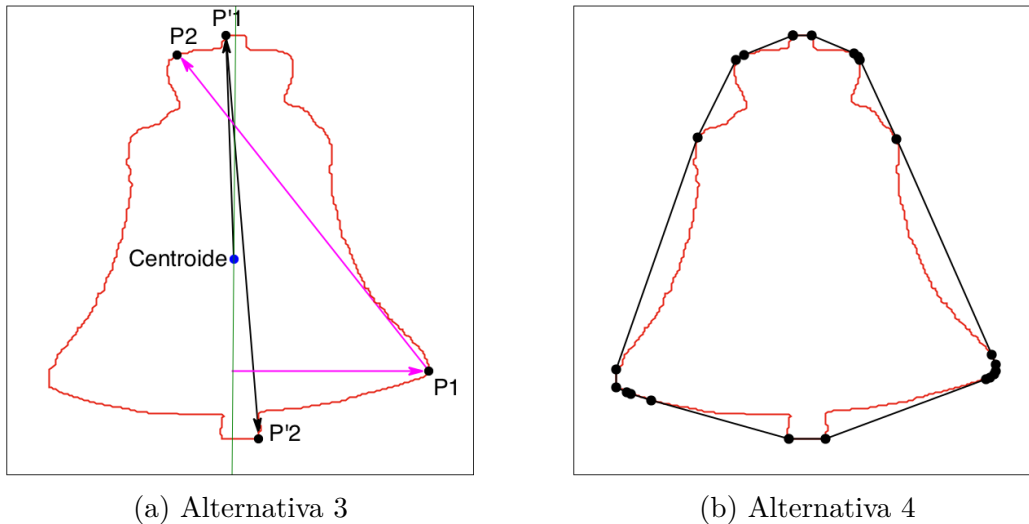


Figura 4.3: Primera etapa: selección de *puntos iniciales*. (c) Alternativa 3. Combinación de las alternativas 1 y 2 (véase la figura 4.2): el punto P_1 es el más alejado del *eje de mínima inercia* y P_2 es el punto más alejado del punto P_1 ; el punto P'_1 es el punto más alejado del centroide y P'_2 es el punto más alejado del punto P'_1 . (d) Alternativa 4. Puntos del *casco convexo*.

4.2.2. Segunda etapa: computación de los valores de significación de los puntos que no son iniciales

La segunda etapa del método calcula los valores de significación de los puntos que no son iniciales. En esta etapa se va a modificar el procedimiento recursivo *calcu-*

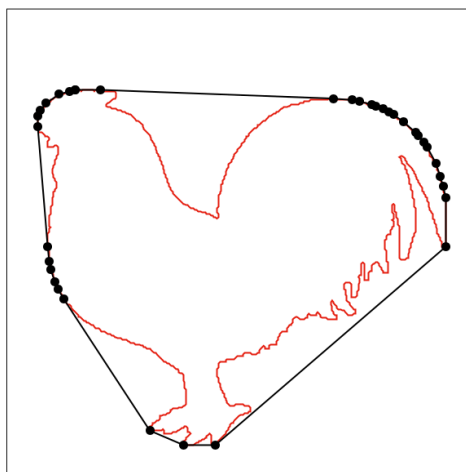


Figura 4.4: Primera etapa: dificultades del uso del *casco convexo* para obtener los *puntos iniciales*. La figura muestra el contorno *chicken-2*, en el que se obtienen demasiados *puntos iniciales* debido a las zonas redondeadas correspondientes a la cabeza, a la cola y al pecho del animal.

lar_significación, descrito en la sección 3.3.2, para tener en cuenta dos criterios:

- **Tipo de medida**

En primer lugar, se va a permitir la elección de la *medida* o *función de distancia* que se desee utilizar. *E_{max}* fue la medida utilizada en el método propuesto en el capítulo 3. El estudio exhaustivo va a permitir comparar el rendimiento de *E_{max}* con el de otras dos medidas: *ISE* y *Longitud*.

- **Simétrico o no simétrico**

En segundo lugar, también se va a modificar el método para que permita elegir si se desea tener en cuenta la simetría del contorno o no. Se introduce esta modificación para comprobar en qué grado influye la simetría en la calidad de la aproximación poligonal generada.

A efectos de clarificar el procedimiento empleado, se describe la nueva versión³:

1. Sea $I = \{P_1, \dots, P_n\}$ el conjunto de *puntos iniciales* obtenido por el método que se describe en la primera etapa del método (véase la sección 4.2.1).

³Las modificaciones aparecen subrayadas o en **negrita**.

2. Ejecutar el procedimiento recursivo $\text{calcular_significación}(P_i, P_{i+1})$. Este proceso se ejecuta para cada pareja de puntos iniciales consecutivos P_i y $P_{(i+1) \bmod n}$ ($\forall i \in \{1, \dots, n\}$):
 - a) Calcular la distancia extrema d_{ext} desde el segmento definido por P y P' a los puntos del contorno que se encuentran entre P y P' . El valor de la distancia extrema d_{ext} se podrá calcular utilizando una de la siguientes medidas:
 - Medida 1.- *E_{max}*
 - Medida 2.- *ISE*
 - Medida 3.- *Longitud*
 - b) Si la distancia extrema d_{ext} es igual a 0.0, se asigna el valor de significación 0.0 a cada punto del contorno que se ubique entre los puntos P y P' . En este caso, el proceso recursivo se detiene.
 - c) Por el contrario, si la distancia extrema d_{ext} no es igual a 0.0 entonces:
 - 1) Seleccionar los puntos del contorno comprendidos entre P y P' en los cuales se haya alcanzado la distancia extrema d_{ext} desde el segmento definido por P y P' .
 - Versión no simétrica: se continúa en el paso 2.
 - Versión simétrica:
 - En la mayoría de los casos, solo existirá un punto, y el proceso continuará en el paso 2.
 - Sin embargo, existe la posibilidad de que exista más de un punto, por ejemplo, si el contorno tiene un lado recto que es paralelo al segmento definido por los puntos P y P' . En esta situación, solamente se seleccionan, entre ellos, los puntos que están más alejados del centroide del contorno. Estos puntos serán considerados como *puntos candidatos* (véase la figura 3.5).
 - Este proceso tiene en cuenta la simetría del contorno.
 - 2) Sean Q_1, Q_2, \dots, Q_m los *puntos candidatos* seleccionados:
 - Asignar el valor de la distancia extrema d_{ext} como *significación* de cada punto Q_i , $i \in \{1, \dots, m\}$.
 - *Ejecutar* el proceso recursivo $\text{calcular_significación}(P, P')$ con cada par de puntos P y Q_1 , Q_1 y Q_2 , \dots , Q_{m-1} y Q_m y, finalmente, Q_m y P' .

El cálculo de la *distancia extrema* d_{ext} dependerá de la *medida* utilizada, porque las alternativas basadas en *E_{max}* o *Longitud* utilizan un proceso de *maximización*, mientras que la alternativa basada en *ISE* utiliza un proceso de *minimización*. A continuación se describe cada alternativa:

Medida 1. *E_{max}*

La medida *E_{max}*, también denominada *error máximo*, calcula la *máxima distancia euclídea* d_{ext} desde el segmento definido por los puntos P_i y P_j hacia los puntos del contorno comprendidos entre dichos puntos P_i y P_j (véase la figura 4.5).

Sea $d(P_k, r)$ la distancia del punto P_k a la recta $r \equiv aX + bY + c = 0$ definida por los puntos P_i y P_j :

$$d(P_k, r) = \frac{|a x_k + b y_k + c|}{\sqrt{a^2 + b^2}} \quad (4.1)$$

- El *error de desviación máxima* del segmento que une P_i y P_j se define como la siguiente distancia *máxima*:

$$E_{max}(P_i, P_j) = \text{máximo}\{d(P_k, r) \mid k \in \{i + 1, \dots, j - 1\}\} \quad (4.2)$$

donde P_k es un punto del contorno comprendido entre P_i y P_j .

Para la medida *E_{max}*, la distancia *extrema* se define como:

$$d_{ext} = E_{max}(P_i, P_j) \quad (4.3)$$

En cada etapa del método se elegirán como *puntos candidatos* a los puntos en los que se alcance la distancia extrema d_{ext} usando la medida *E_{max}*, para así obtener la máxima reducción del error de ajuste de la aproximación poligonal al contorno.

Medida 2. *I_{SE}*

La medida *I_{SE}* (*Integral Squared Error*), también conocida como *error cuadrático integral*, calcula la *suma de los cuadrados de las distancias* desde el segmento definido por los puntos P_i y P_j hacia los puntos del contorno comprendidos entre dichos puntos P_i y P_j :

$$I_{SE}(P_i, P_j) = \sum_{k=i+1}^{j-1} d^2(P_k, r) \quad (4.4)$$

donde $d(P_k, r)$ es la distancia del punto P_k a la recta r (véase la expresión 4.1).

Para la medida *I_{SE}*, la distancia *extrema* se define como la siguiente distancia *mínima*:

$$d_{ext} = \text{mínimo}\{I_{SE}(P_i, P_k) + I_{SE}(P_k, P_j) \mid k \in \{i + 1, \dots, j - 1\}\} \quad (4.5)$$

donde P_k es un punto del contorno comprendido entre P_i y P_j .

En cada etapa del método, se elegirán los puntos en los que se alcance la distancia extrema d_{ext} , que minimiza el error de ajuste de la aproximación poligonal al contorno usando la medida *I_{SE}*. Véase la figura 4.6.

Medida 3. *Longitud*

La medida número 3 va a tener en cuenta el *incremento de la longitud* de la aproximación poligonal al añadir un nuevo punto.

Dados dos puntos $P_i = (x_i, y_i)$ y $P_j = (x_j, y_j)$, la longitud del segmento que los une se calcula mediante su distancia euclídea:

$$d(P_i, P_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (4.6)$$

Al añadir un nuevo punto P_k a la aproximación poligonal, se produce un incremento de su longitud o perímetro que se puede calcular mediante la siguiente expresión:

$$\Delta L(i, k, j) = d(P_i, P_k) + d(P_k, P_j) - d(P_i, P_j) \quad (4.7)$$

donde P_k es un punto del contorno comprendido entre P_i y P_j .

Para la medida *Longitud*, la distancia *extrema* se define como la distancia que *maximiza* el *incremento de la longitud* de la aproximación poligonal.

$$d_{ext} = \text{máximo}\{L(i, j, k) \mid k \in \{i + 1, \dots, j - 1\}\} \quad (4.8)$$

En cada etapa del método se elegirán los puntos que alcancen la distancia extrema d_{ext} , que *maximiza* el incremento de la longitud de la aproximación poligonal resultante.

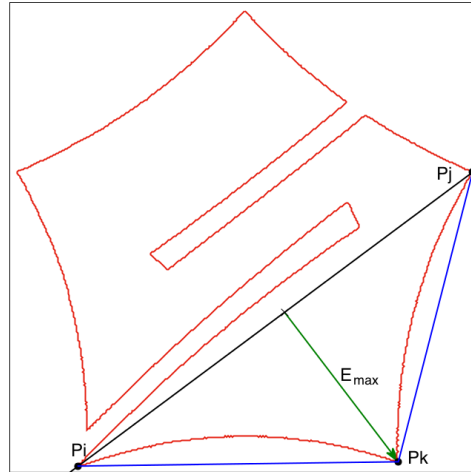


Figura 4.5: Segunda etapa: cálculo de la medida E_{max} . Se selecciona el punto P_k , que genera el error máximo E_{max} (contorno *device6-9*).

La complejidad computacional de esta etapa es $\mathcal{O}(n \log(n))$ en el mejor de los casos y $\mathcal{O}(n^2)$ en el peor escenario [90], si todos los *break points* son dominantes.

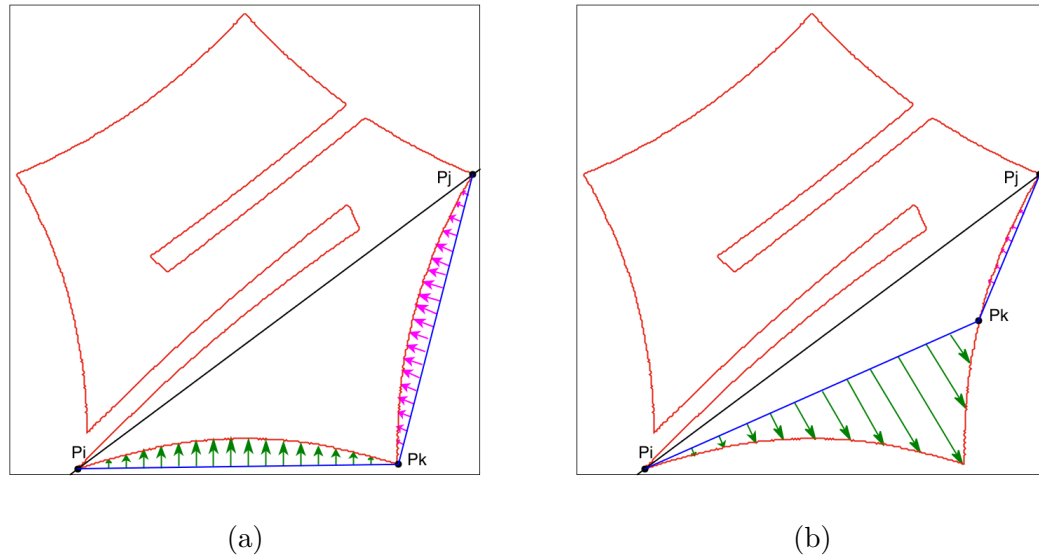


Figura 4.6: Segunda etapa: cálculo de la medida *ISE*. La figura (a) muestra la detección del punto P_k en el que se **minimiza** la medida *ISE* al calcular las distancias de los segmentos definidos por los puntos P_i y P_k y P_k y P_j a los puntos del contorno *device6-9*. La figura (b) muestra otro punto P_k donde no se alcanza el mínimo para la medida *ISE*.

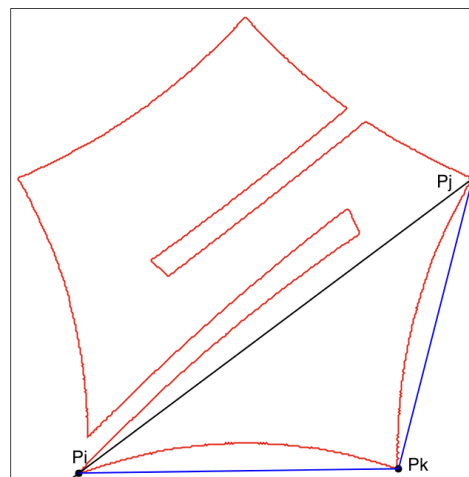


Figura 4.7: Segunda etapa: cálculo de la medida *Longitud*. La figura muestra el punto P_k , comprendido entre los puntos P_i y P_j del contorno *device6-9*, que **maximiza** el incremento de longitud de la aproximación poligonal: $\Delta L(i, k, j) = d(P_i, P_k) + d(P_k, P_j) - d(P_i, P_j)$.

4.2.3. Tercera etapa: computación de los valores de significación de los puntos iniciales

El objetivo de esta etapa es garantizar que los puntos iniciales tengan el *valor máximo de significación*. En el método propuesto en el capítulo 3, se utilizó un criterio basado en la significación máxima de los puntos *no iniciales* (Sig_{max}) y en la máxima distancia desde el *centroide* hasta los puntos del contorno (véase la sección 3.3.3). En este capítulo se propone una alternativa adicional para que también se tenga en cuenta la máxima distancia del eje de mínima inercia a los puntos del contorno.

- Computar el valor máximo de significación de los puntos que no son iniciales: Sig_{max} .
- Si Sig_{max} es igual a 0.0, entonces el valor de significación de los puntos *iniciales* será 1.0. Esta situación ocurre con contornos artificiales en los que los puntos iniciales definen una aproximación poligonal exacta.
- Por el contrario, si Sig_{max} no es igual a 0.0, se aplicará uno de los siguientes métodos:
 1. El valor de significación de los puntos iniciales será el máximo entre Sig_{max} y d , donde d es la distancia máxima desde el *eje de mínima inercia* hasta los puntos del contorno.
 2. El valor de significación de los puntos iniciales será el máximo entre Sig_{max} y d , donde d es la distancia máxima desde el *centroide* hasta los puntos del contorno⁴.

La complejidad computacional de esta etapa es $\mathcal{O}(n)$, puesto que el cálculo del valor máximo de significación de los puntos del contorno se realiza mediante un proceso lineal.

4.2.4. Séptima etapa: borrado de puntos dominantes superfluos generados por el casco convexo (opcional)

Esta séptima etapa de borrado de puntos dominantes *superfluos* es opcional y solamente se aplicará si, en la primera etapa, se utilizó el *casco convexo* del contorno para seleccionar los puntos iniciales (véase la sección 4.2.1). La utilización del *casco convexo* presenta el inconveniente de que puede provocar que el método genere demasiados puntos dominantes en contornos con partes o zonas redondeadas (véase la figura 4.8.) Muchos de estos puntos dominantes son *superfluos* por que están *casi alineados* y, por tanto, pueden ser eliminados.

El proceso de supresión de puntos dominantes *superfluos* es el siguiente:

⁴Esta alternativa es la que se utilizó en el método original propuesto en el capítulo 3.

- Sean P_{i-1}, P_i y P_{i+1} tres puntos dominantes *consecutivos* de la aproximación poligonal.
- El punto P_i se puede borrar de la lista de puntos dominantes si se cumple alguno de los siguientes criterios:
 1. *Primer criterio*: minimización de E_{max} :

$$\begin{aligned} & (E_{max}(P_{i-1}, P_{i+1}) < E_{max}(P_{i-1}, P_i)) \\ \wedge & (E_{max}(P_{i-1}, P_{i+1}) < E_{max}(P_i, P_{i+1})) \end{aligned} \quad (4.9)$$

Siendo $E_{max}(\cdot, \cdot)$ el *error de desviación máxima* (véase la expresión 4.2).

El punto P_i será borrado si el error de desviación máxima del lado definido por los puntos P_{i-1} y P_{i+1} es menor que los errores de desviación máxima de los otros dos lados definidos por P_{i-1} y P_i , y P_i y P_{i+1} .

2. *Segundo criterio*: minimización de ISE :

$$ISE(P_{i-1}, P_{i+1}) < ISE(P_{i-1}, P_i) + ISE(P_i, P_{i+1}) \quad (4.10)$$

Siendo $ISE(\cdot, \cdot)$ el *error cuadrático integral* (véase la expresión 4.4).

El punto P_i será borrado si el *error cuadrático integral* del lado definido por los puntos P_{i-1} y P_{i+1} es menor que la suma de los *errores cuadráticos integrales* de los otros dos lados definidos por P_{i-1} y P_i , y P_i y P_{i+1} .

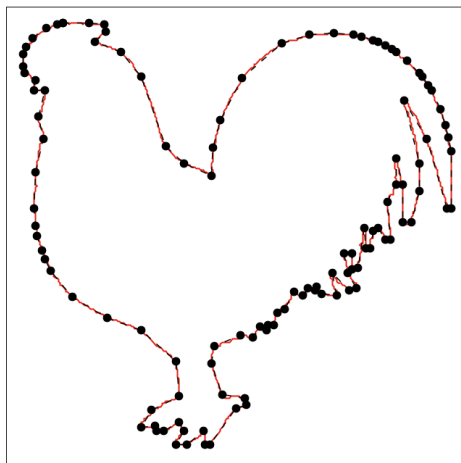
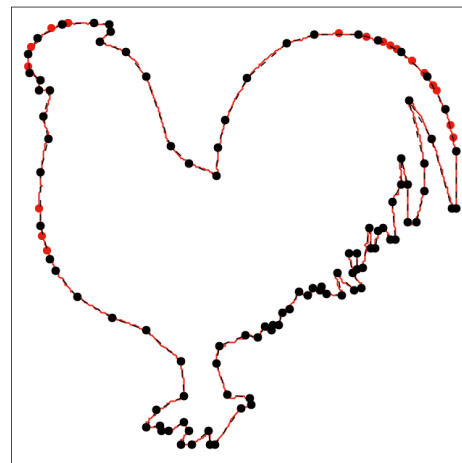
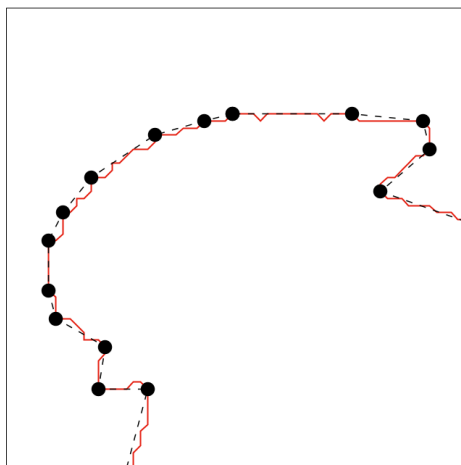
- La verificación de cualquiera de los dos criterios provoca que la supresión del punto P_i mejore el ajuste de la aproximación poligonal al contorno.
- Este proceso de supresión se repite hasta que ya no se pueda eliminar ningún punto dominante o hasta que la aproximación tenga solamente tres vértices (triángulo).

Una vez que se ha completado el proceso de borrado, los puntos dominantes restantes son los *vértices* de la aproximación poligonal (véase la figura 4.8). La complejidad computacional de esta etapa es lineal: $\mathcal{O}(n)$.

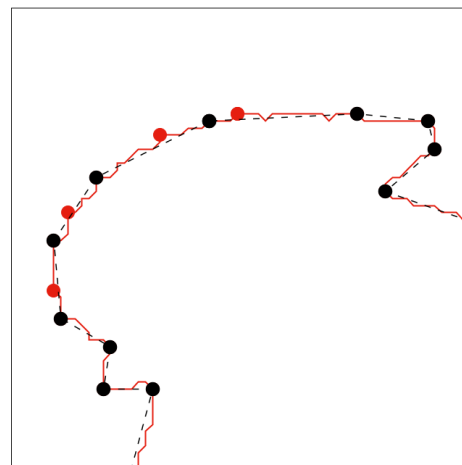
4.3. Alternativas basadas en la estrategia *merge*

En la sección 4.2 se han propuesto modificaciones, basadas en la estrategia *split*, del método de generación de aproximaciones poligonales descrito en el capítulo 3. En esta sección, se proponen alternativas basadas en la estrategia *merge* de fusión de puntos (véase la figura 4.9):

Las alternativas contempladas son:

(a) Aproximación de *chicken-2*(b) Puntos superfluos de *chicken-2*

(c) Detalle de la cabeza de la figura



(d) Puntos superfluos en la cabeza de la figura

Figura 4.8: Séptima etapa: borrado de puntos dominantes superfluos generados por el casco convexo: (a) Aproximación de *chicken-2*; (b) Puntos *superfluos* de *chicken-2* (puntos resaltados en rojo); (c) Detalle de la cabeza de la figura *chicken-2*; (d) Puntos *superfluos* de la cabeza de la figura *chicken-2* (puntos resaltados en rojo).

1. *Selección de los puntos iniciales del contorno*: se utiliza el código de cadena para detectar los puntos de ruptura (*break points*); es decir, los puntos del contorno en los que hay un cambio de dirección. Estos puntos serán considerados como los *puntos iniciales* del método (véase la sección 4.3.1).
2. *Supresión de puntos y cálculo de su significación*: los *puntos iniciales* se eliminan uno a uno, de forma que, en cada etapa, su significación o *error de desviación* a la aproximación poligonal actual del contorno alcance un valor *mínimo* para la medida de error utilizada (*E_{max}*, *ISE* y *Longitud*). Véase la sección 4.3.2.
3. *Cálculo de la significación acumulada*: se calcula la significación acumulada de cada punto del contorno teniendo en cuenta el orden en el que fueron suprimidos (véase la sección 4.3.3).
4. *Generación de la curva de significación acumulada y normalizada*: se normalizan los valores de *significación normalizada* para generar esta nueva curva. Este proceso es descrito en la sección 3.3.4.
5. *Búsqueda del umbral de los valores de significación acumulados y normalizados*: se calcula el umbral de significación con las mismas alternativas que se usaron en el método original (véase la sección 3.3.5).
6. *Umbralización de la curva de significación acumulada y normalizada*: el proceso utilizado para umbralizar los valores de significación es idéntico al que se usó en el método original (véase la sección 3.3.6).

En las secciones 4.3.1-4.3.3 se detallan los cambios introducidos en las etapas primera, segunda y tercera de este método respecto al método original.

4.3.1. Primera etapa: selección de los puntos iniciales

El proceso es el siguiente:

1. Parametrizar el contorno para que el primer punto sea el que esté más alejado del centroide.
2. Calcular el *código de la cadena* de los puntos del contorno.
3. Marcar como *puntos iniciales* a los *break points*; es decir, a los puntos en los que se produce un cambio de dirección al recorrer el contorno.
4. Eliminar todos los puntos del contorno que no son *break points*, porque forman parte de tramos rectos o alineados. Estos puntos tendrán un valor de significación igual a 0.0 y su orden de borrado es 0. Este orden de borrado hace referencia al instante en el que se borran los puntos del contorno durante el proceso y se ha incrementará a lo largo de las diferentes fases que lo componen.

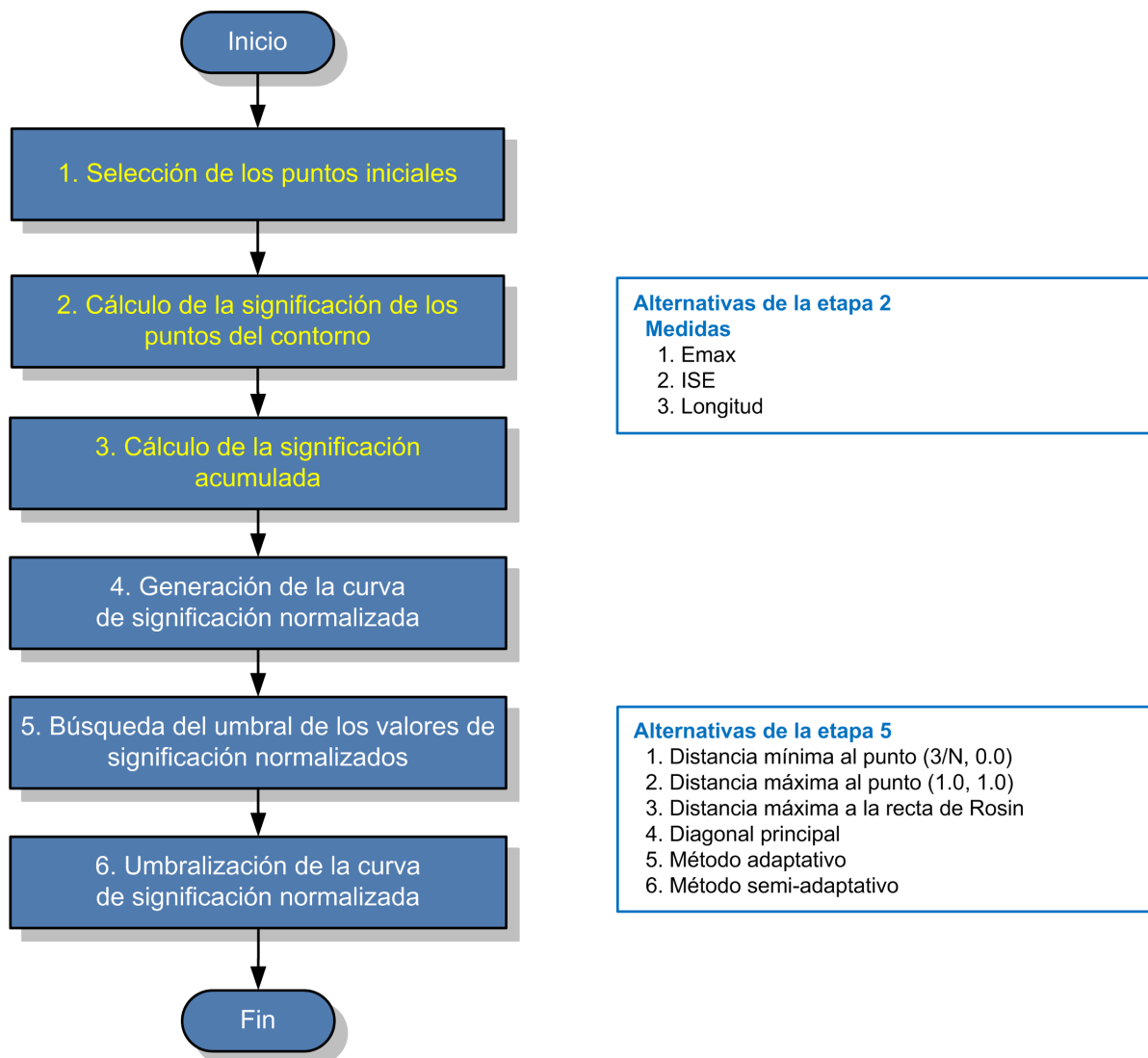


Figura 4.9: Esquema del método de la estrategia *merge*. Se resaltan en amarillo las etapas que han sufrido cambios respecto del método original.

La complejidad computacional de esta etapa es lineal $\mathcal{O}(n)$, puesto que se recorre el contorno una única vez.

4.3.2. Segunda etapa: cálculo de la significación de los puntos del contorno

La estrategia *merge* es usada para eliminar, de forma paulatina, el *punto inicial* del contorno en el que se alcance el valor *mínimo* de significación (véase la sección 2.1). Al igual que se estableció en la alternativa basada en la estrategia *split* (véase la sección 4.2.2), se va a permitir la elección de la *medida* o *función de distancia* que se desee utilizar para calcular la significación de los puntos del contorno. Se podrán utilizar las *medidas* que fueron propuestas para la modificación del método basada en la estrategia *split* (véase la sección 4.2.2); es decir: *E_{max}*, *ISE* y *Longitud*.

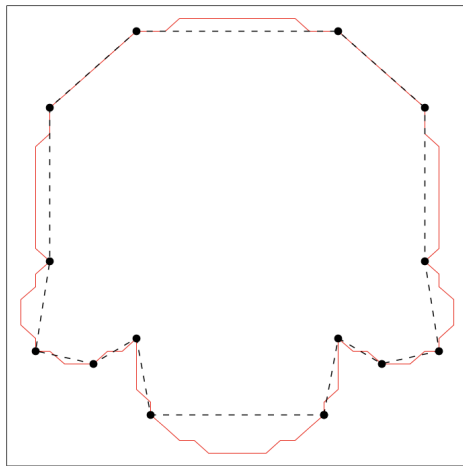
La modificación basada en la estrategia *merge* no va a tener en cuenta la simetría del contorno, debido a que es necesaria una fase posterior de *cálculo de significación acumulada* (véase la sección 4.3.3) que utiliza el número de orden de borrado de los *puntos iniciales* del contorno. Se debe tener en cuenta que, en cada etapa del método, se va a eliminar el punto con *menor* significación y se le va a asignar un número de orden de borrado. Si hubiera dos o más puntos en los que se alcanzara el valor *mínimo* entonces habría dos posibilidades:

1. Eliminarlos simultáneamente y asignarles el mismo número de orden de borrado.
2. Eliminar solamente uno de ellos (por ejemplo, el más cercano al centroide) y asignarle su número de orden de borrado.

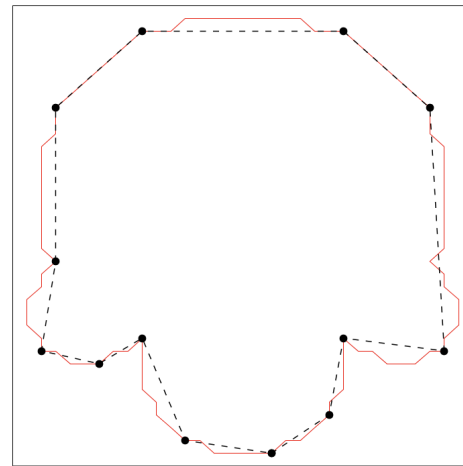
Sin embargo, la primera opción no se puede utilizar porque la siguiente etapa requiere que todos los *puntos iniciales* tengan un número orden de borrado diferente para poder calcular la *significación acumulada* de los puntos. La fase de *cálculo de significación acumulada* es necesaria porque se ha comprobado que no se obtienen buenos resultados si no se utiliza (véase la figura 4.10).

El procedimiento para calcular la significación de los puntos del contorno utilizando la modificación basada en la estrategia *merge* es el siguiente:

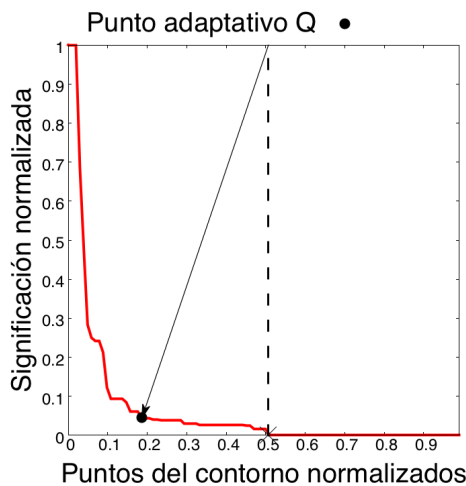
1. Obtención de los *puntos iniciales*:
 - Sea $I = \{P_1, \dots, P_n\}$ el conjunto de *puntos iniciales*, que se obtiene en la etapa anterior del método (véase la sección 4.3.1).



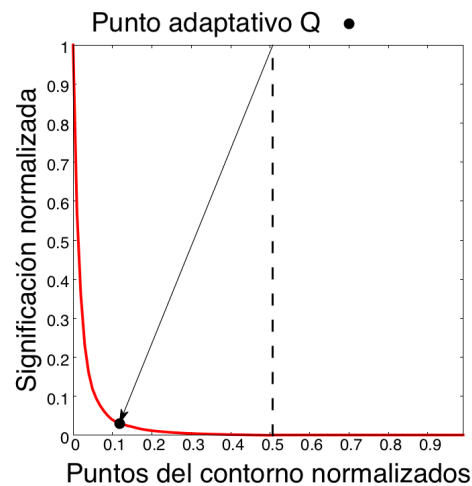
(a) Aproximación del contorno *semicircles* con significación *sin acumular* (DP = 22, $WE_2 = 1,6273$)



(b) Aproximación poligonal *semicircles* con significación *acumulada* (DP = 13, $WE_2 = 0,8026$)



(c) Curva de significación normalizada *sin acumular*



(d) Curva de significación normalizada *acumulada*

Figura 4.10: Segunda etapa: cálculo de la significación de los puntos del contorno *semicircles* (102 puntos). Problemas de simetría al calcular la significación acumulada: (a) Aproximación poligonal simétrica obtenida al usar la significación *sin acumular*; (b) Aproximación poligonal no simétrica obtenida al usar la significación *acumulada*; (c) Curva de significación normalizada *sin acumular*; (d) Curva de significación *acumulada* y normalizada.

2. Cálculo de la significación inicial de los *puntos iniciales*⁵:

- Para cada punto $P_i \in I$,
 - Calcular la *distancia mínima* d_{min_i} desde el punto P_i al segmento definido por $P_{(i-1) \bmod n}$ y $P_{(i+1) \bmod n}$, usando una de las siguientes *medidas*:
 - Medida 1.- *E_{max}* (véase la expresión 4.2).
 - Medida 2.- *ISE* (véase la expresión 4.4).
 - Medida 3.- *Longitud* (véase la expresión 4.7).
 - El valor de d_{min_i} representa la significación *inicial* y *actual* del punto P_i .

3. Inicializar el número de orden de borrado de los *puntos iniciales*:

$$orden_borrado \leftarrow 1$$

4. Supresión paulatina del *punto inicial* con *menor* significación actual:

- Mientras el conjunto I tenga más de tres puntos⁶:
 - a) Seleccionar el punto P_k con menor valor de significación d_{min_k} .
Si hay varios puntos en los que se alcance el valor mínimo, se elige, entre ellos, el punto más cercano al centroide. Si la distancia mínima se alcanzara en más de un punto, se elegiría el punto con menor número de parametrización.
 - b) Eliminar el punto P_k del conjunto I :
 $I \leftarrow I - \{P_k\}$
 - c) Asignar el número de orden de borrado al punto P_k :
 $P_k.orden_borrado \leftarrow orden_borrado$
 - d) Incrementar el número de orden de borrado:
 $orden_borrado \leftarrow orden_borrado + 1$
 - e) Actualizar los valores de *significación actual* de los puntos anterior y posterior a P_k en el conjunto I , es decir, $P_{(k-1) \bmod n}$ y $P_{(k+1) \bmod n}$.

5. Cálculo de la significación de los tres últimos puntos:

- Sea $I = \{P, P', P''\}$ el conjunto resultante del paso anterior que está compuesto por los tres últimos puntos.
- Asignar a P , P' y P'' un valor de significación igual a la máxima distancia al centroide del contorno⁷.

⁵En la primera fase, descrita en la sección 4.3.1, se ha asignado una significación igual a 0.0 y un número de borrado igual a 0 a los puntos que *no* son iniciales o que están alineados.

⁶Se ha elegido el valor *tres* puesto que representa a la aproximación poligonal con el menor número de lados: el *triángulo*.

⁷La máxima distancia al centroide la proporciona el primer punto del contorno, porque ha sido parametrizado de forma que el primer punto sea el que esté más alejado del centroide (véase la sección 4.3.1).

- Asignar el número de orden de borrado a los puntos P , P' y P'' .

$$P.\text{orden_borrado} \leftarrow \text{orden_borrado}$$

$$P'.\text{orden_borrado} \leftarrow \text{orden_borrado} + 1$$

$$P''.\text{orden_borrado} \leftarrow \text{orden_borrado} + 2$$

La complejidad computacional de esta etapa es $\mathcal{O}(n \log(n))$.

4.3.3. Tercera etapa: cálculo de la significación acumulada

En la tercera etapa se calcula la significación *acumulada* de los puntos del contorno. Para este cálculo se tendrá en cuenta el orden en el que los puntos fueron borrados. El proceso a seguir es el siguiente:

1. Ordenar de forma decreciente los puntos del contorno según su número de orden de borrado⁸.
2. Calcular la significación acumulada de los puntos del contorno.

Si $Sig(P_i)$ es la significación del punto P_i , entonces la significación acumulada de P se representa por $Sig_{ac}(P_i)$ y se calcula de la siguiente forma:

$$\begin{aligned} Sig_{ac}(P_n) &= Sig(P_n) \\ Sig_{ac}(P_i) &= Sig(P_i) + \sum_{j=n}^{i+1} Sig_{ac}(P_j) \quad \forall(j > i) \end{aligned} \tag{4.11}$$

Se debe indicar que el cálculo de la significación acumulada está justificado, porque la fase posterior de búsqueda del umbral requiere que la curva de significación normalizada sea estrictamente decreciente y con una *esquina* muy pronunciada. Si no se calcula la significación acumulada, la curva de significación puede no ser estrictamente decreciente, debido a la aparición de *máximos locales* (véase la figura 4.10 (c)) o puede carecer de una *esquina* pronunciada (véase la figura 4.11 (c)).

El cálculo de la significación acumulada permite generar una curva estrictamente decreciente y con una *esquina* muy pronunciada que facilita el proceso de búsqueda del *umbral* heurístico. En las figuras 4.11 (a) y (b), se muestran las aproximaciones poligonales del contorno *device6-9* si no se aplica el cálculo de significación *acumulada* y si se aplica, respectivamente. Al acumular la significación, el ajuste de la aproximación poligonal mejora de forma notable (véanse las figuras 4.11 (c) y (d) y la expresión 4.11).

⁸Para realizar la ordenación se utiliza el algoritmo de ordenación *Quicksort*, que proporciona una complejidad computacional de $\mathcal{O}(n \log(n))$ en el mejor caso (en el caso promedio) y de $\mathcal{O}(n^2)$ en el peor de los casos.

Como contrapartida, el proceso de *acumulación* provoca que el método no sea simétrico, debido al proceso de ordenamiento utilizado.

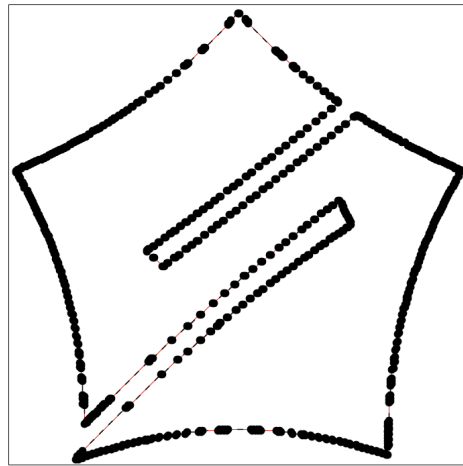
4.4. Evaluación de los métodos y análisis de los resultados

En las secciones 4.2 y 4.3 se han analizado las parametrizaciones del método original basadas en las estrategias *split* y *merge*, respectivamente. Cada estudio define un conjunto de versiones que dependen de las alternativas que se elijan en algunas de sus etapas. Con el objetivo de evaluar y comparar dichas versiones, se han desarrollado dos experimentos:

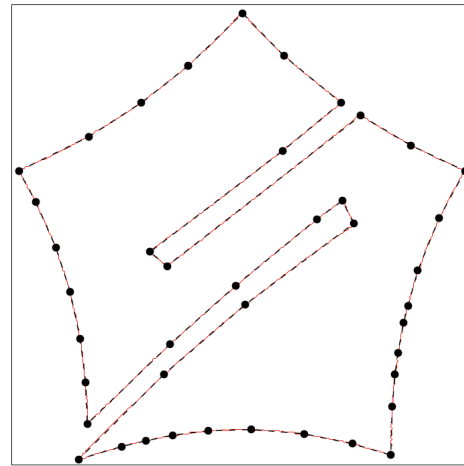
- Primer experimento: selección de la mejor versión del método basado en la estrategia *split*.
 - La combinación de las alternativas de las diferentes etapas permite definir 288 versiones diferentes del método.
 - La descripción de este primer experimento se puede consultar en la sección 4.4.1.
- Segundo experimento: selección de la mejor versión del método basado en la estrategia *merge*.
 - La combinación de las alternativas de las diferentes etapas permiten definir 18 versiones del método.
 - La descripción de este segundo experimento se puede consultar en la sección 4.4.2.

En los dos experimentos se ha vuelto a utilizar la misma base de datos *MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)* [42, 87, 92], que ya fue utilizada para evaluar el rendimiento del método original propuesto en la sección 3.3.

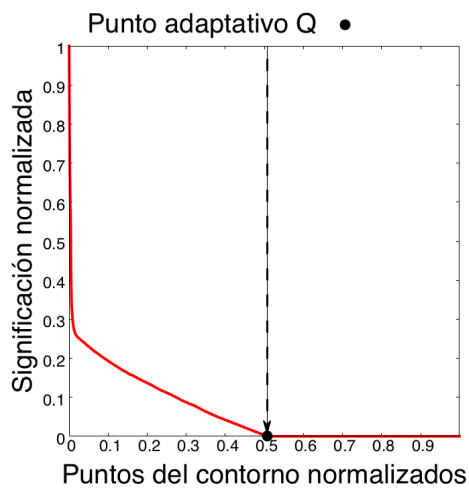
Para evaluar la calidad de las aproximaciones de cada experimento, se han utilizado las siguientes medidas: número de puntos dominantes (*DP*), ratio de compresión (*CR*), error cuadrático integral (*ISE*), error máximo (E_∞) y la suma ponderada del error cuadrático (WE_2) (véase la sección 2.2). También se ha utilizado la medida combinada $\mu WE_2 + \sigma WE_2$ para evaluar la estabilidad del rendimiento de cada una de las versiones de los métodos. Por último, se ha calculado la medida de comparación *Index WE₂* (véanse las tablas 4.2 y 4.4).



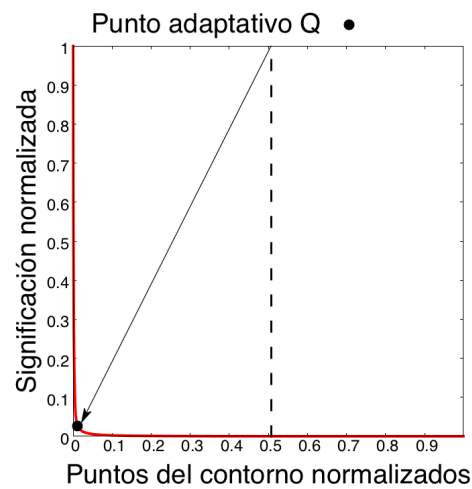
(a) Aproximación poligonal con significación *sin acumular*
 $DP = 806$, $WE_2 = 2,087$



(b) Aproximación poligonal con significación *acumulada*
 $DP = 42$, $WE_2 = 0,1257$



(c) Curva de significación *sin acumular* y normalizada



(d) Curva de significación *acumulada* y normalizada

Figura 4.11: Tercera etapa: cálculo de la significación acumulada para el contorno *device6-9* (1590 puntos): (a) aproximación poligonal obtenida con la significación *sin acumular*; (b) aproximación poligonal obtenida con la significación *acumulada*; (c) curva de significación *sin acumular* y normalizada; (d) curva de significación *acumulada* y normalizada

4.4.1. Primer experimento: selección de la mejor versión del método basado en la estrategia *split*

El estudio basado en la estrategia *split* del método original posee alternativas en algunas de sus etapas (véanse la figura 4.1 y la tabla 4.1). Dado que hay cuatro (4) alternativas para la primera etapa (*selección de puntos iniciales*), seis (6) alternativas para la segunda etapa (*cálculo de la significación de los puntos no iniciales*), dos (2) alternativas para la tercera etapa (*cálculo de la significación de los puntos iniciales*) y seis (6) alternativas para la quinta etapa (*métodos de umbralización*), se pueden definir un total de $4 \times 6 \times 2 \times 6 = 288$ versiones diferentes del método basado en la estrategia *split*.

El objetivo de este experimento es seleccionar la mejor versión del método *split* a partir de la combinación de sus alternativas. Para identificar cada versión del método, se va a utilizar la siguiente notación: $n_1 - n_2 - n_3 - n_4$, donde (véanse la figura 4.1 y la tabla 4.1):

- $n_1 \in \{1, 2, 3, 4\}$ es una de las alternativas de *selección de puntos iniciales* (véase la sección 4.2.1).
- $n_2 \in \{1, 2, 3, 4, 5, 6\}$ es una de las alternativas de *cálculo de la significación de los puntos no iniciales* (véase la sección 4.2.2).
- $n_3 \in \{1, 2\}$ es una de las alternativas de *cálculo de la significación de los puntos iniciales* (véase la sección 4.2.3).
- $n_4 \in \{1, 2, 3, 4, 5, 6\}$ es una de las alternativas de los *métodos de umbralización* (véase la sección 3.3.5).

Por ejemplo, el identificador 2-1-2-5 representa a la versión que utiliza:

- La alternativa n° 2 para seleccionar los puntos iniciales; es decir, la máxima distancia al centroide.
- La alternativa n° 1 para utilizar la medida *E_{max}* y la simetría al calcular la significación de los puntos no iniciales.
- La alternativa n° 2 para usar la máxima distancia al centroide al calcular la significación de los puntos iniciales.
- La alternativa n° 5 para aplicar el método de umbralización adaptativo.

Esta versión es la que se corresponde con el método original propuesto en la sección 3.3.

Tabla 4.1: Tabla de alternativas del método *split*

Etapa 1.- Iniciales	Etapa 2.- Medida y simetría
1. Eje de mínima inercia	1. E_{max} simétrica
2. Centroide	2. E_{max} no simétrica
3. Combinación de alternativas 1 y 2	3. ISE simétrica
4. Casco convexo	4. ISE no simétrica
	5. Longitud simétrica
	6. Longitud no simétrica
Etapa 3.- Significación iniciales	Etapa 5.-Umbralización
1. Distancia eje de mín. inercia	1. Distancia mín. a (3/N, 0.0)
2. Distancia al centroide	2. Distancia máx. a (1.0, 1.0)
	3. Distancia máx. a recta de Rosin
	4. Diagonal principal
	5. Método adaptativo
	6. Método semi-adaptativo

Nota: la combinación de las alternativas permite definir un total de **288 versiones** diferentes del método.

La evaluación cuantitativa las versiones se muestra en la tabla 4.2 y en la figura 4.12, que muestran el valor medio de las medidas de calidad, la desviación típica de la medida WE_2 , la medida combinada $\mu WE_2 + \sigma WE_2$ y el atributo $Index WE_2$. La tabla 4.2 solo muestra los resultados de las 20 mejores versiones del método, que están ordenadas de forma creciente según el valor de $Index WE_2$ (última columna de la tabla). Se añade también la versión 2-1-2-5, que se encuentra en la posición número 25 de la tabla. Un análisis más detallado permite realizar las siguientes observaciones:

- **Posición 1:** versión [4-2-1-6](#)
 - Las alternativas de esta versión son las siguientes:
 - **Selección de iniciales:** 4.- Casco convexo
 - **Medida y simetría:** 2.- E_{max} no simétrica
 - **Significación de iniciales:** 1.- Distancia máxima al centroide
 - **Umbralización:** 6.- Método semi-adaptativo

- Esta versión ocupa la primera posición porque ha obtenido el mejor valor combinado $\mu WE_2 + \sigma WE_2 = 3,4295$, lo que la convierte en la versión más estable entre las 288 versiones analizadas. Si se analizan de forma independiente la media y la desviación típica, esta versión obtiene el tercer mejor valor para la media $\mu WE_2 = 1,5381$ y el mejor valor para la desviación típica $\sigma WE_2 = 1,8915$.
 - Esta versión utiliza la variante de selección de puntos iniciales basada en el *casco convexo*, por lo que presenta el inconveniente de tener que aplicar la séptima etapa de supresión de puntos superfluos (véanse la sección 4.2.4 y la figura 4.8). Analizando los 1400 contornos de la base de datos de pruebas, esta versión genera aproximaciones poligonales que tienen un promedio de *63 puntos dominantes*, sin contar con la etapa opcional de supresión de puntos superfluos. Mediante el uso de la etapa adicional, se consigue eliminar un promedio de *15 puntos superfluos*, lo que reduce en un 23 % la cantidad de puntos dominantes promedio de la aproximación poligonal resultante.
- **Posición 3:** versión *4-1-1-5*
- Las alternativas de esta versión son las siguientes:
 - **Selección de iniciales:** 4.- Casco convexo
 - **Medida y simetría:** 1.- Emax *simétrica*
 - **Significación de iniciales:** 1.- Distancia máxima al centroide
 - **Umbralización:** 5.- Método adaptativo
 - Entre las 288 versiones analizadas, esta versión es la que ha obtenido el mejor valor promedio $\mu WE_2 = 1,4592$; sin embargo, su desviación típica es el quinto valor más pequeño ($\sigma WE_2 = 1,9785$), lo que provoca que la medida combinada $\mu WE_2 + \sigma WE_2 = 3,4377$ coloque a esta versión en la tercera posición de la tabla, siendo su *Index WE₂* = 1,0024.
 - Esta versión también utiliza la variante de selección de puntos iniciales basada en el *casco convexo*, por lo que presenta el inconveniente de tener que suprimir los puntos superfluos, como ya se ha indicado en la versión anterior. Analizando los 1400 contornos de la base de datos de pruebas, esta versión genera aproximaciones poligonales que tienen un promedio de *69 puntos dominantes*, sin contar con la etapa opcional de supresión de puntos superfluos. Mediante el uso de la etapa adicional, se consigue eliminar un promedio de *12 puntos superfluos*, lo que reduce en un 17 % la cantidad de puntos dominantes promedio de la aproximación poligonal resultante.
- **Posición 13:** versión *2-1-1-6*
- Las alternativas de esta versión son las siguientes:
 - **Selección de iniciales:** 2.- Distancia al centroide
 - **Medida y simetría:** 1.- Emax *simétrica*
 - **Significación de iniciales:** 1.- Distancia máxima al centroide
 - **Umbralización:** 6.- Método semi-adaptativo

- Esta versión es la mejor entre todas las que **no** utilizan el casco convexo para seleccionar los puntos iniciales, evitando tener que aplicar la fase final de borrado de puntos superfluos (véase la sección 4.2.4). El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 4,0605 y su *Index WE₂* = 1,1840.
- **Posición 25:** versión 2-1-2-5
 - Esta es la versión que se corresponde con el método propuesto en la sección 3.3. Sus alternativas son las siguientes:
 - **Selección de iniciales:** 2.- Máxima distancia al centroide.
 - **Medida y simetría:** 1.- Emax *simétrica*
 - **Significación de iniciales:** 2.- Distancia máxima al centroide
 - **Umbralización:** 5.- Método adaptativo
 - El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 4,1611 y su *Index WE₂* = 1,2133.

Las figuras 4.13 y 4.14 permiten realizar una evaluación cualitativa de las aproximaciones poligonales generadas por las cuatro alternativas estudiadas del método *split* (4-2-1-6, 4-1-1-5, 2-1-1-6 y 2-1-2-5) para dos contornos de la base de datos de pruebas MPEG-7.

Un análisis de los resultados de este primer experimento permite extraer las siguientes conclusiones:

- La tabla 4.2 indica que el estudio de la estrategia *split* ha permitido obtener 24 versiones (de las 288 posibles) que tienen un mejor rendimiento que el método original propuesto en la sección 3.3.
- La versión 4-2-1-6, basada en el casco convexo para seleccionar los puntos iniciales, es la que ha obtenido los mejores resultados si se tiene en cuenta la medida combinada $\mu WE_2 + \sigma WE_2$. Esta versión requiere de una fase final para suprimir puntos superfluos casi alineados (véase la sección 4.2.4).
- La versión 2-1-1-6 es la que ha obtenido los mejores resultados entre todas las que **no** utilizan el casco convexo para seleccionar los puntos iniciales. Además, esta versión también supera al método original propuesto en la sección 3.3.

Sin embargo, y a pesar de que la versión 4-2-1-6 es la que obtiene mejores resultados, la versión 4-1-1-6 obtiene resultados muy similares en la medida combinada $\mu WE_2 + \sigma WE_2$. Esto implica que la versión candidata *split* estaría compuesta por las siguientes alternativas:

- **Selección de iniciales:** 4.- Casco convexo
- **Medida y simetría:** 1.- Emax *simétrica* o 2.- Emax *no simétrica*

- **Significación de iniciales:** 1.- Distancia máxima al centroide
- **Umbralización:** 6.- Método semi-adaptativo

Las versiones basadas en la estrategia *split* que han obtenido los mejores resultados son *4-2-1-6* y *4-1-1-6*. Ambas versiones utilizan el casco convexo como método de selección de puntos iniciales. En la segunda etapa, la versión *4-2-1-6* utiliza la medida *E_{max}* de forma no simétrica para calcular la significación de los puntos del contorno. En cambio, la versión *4-1-1-6* sí tiene en cuenta la simetría. En la tercera etapa, ambas versiones calculan la significación de puntos iniciales a partir de la máxima distancia al centroide. Por último, en la etapa de umbralización, ambas versiones utilizan el método *semi-adaptativo*.

En resumen, este primer experimento ha mostrado que el estudio de la estrategia *split* permite definir nuevas versiones del método que mejoran el rendimiento del método original propuesto en el capítulo 3.

4.4.2. Segundo experimento: selección de la mejor versión del método basado en la estrategia *merge*

El estudio de la estrategia *merge* (véanse la figura 4.9 y la tabla 4.3) posee alternativas en dos de sus etapas: tres (3) alternativas en la segunda etapa de *cálculo de la significación de los puntos del contorno* y seis (6) alternativas en la quinta etapa de *búsqueda del umbral* de los valores de significación acumulados y normalizados. La combinación de estas alternativas permite definir un total de $3 \times 6 = 18$ versiones diferentes del método basado en la estrategia *merge*.

El objetivo de este experimento es seleccionar la mejor versión del método *merge* a partir de la combinación de sus alternativas. Para identificar cada versión del método, se va a utilizar la siguiente notación: $n_1 - n_2$, donde (véanse la figura 4.9 y la tabla 4.3):

- $n_1 \in \{1, 2, 3\}$ es una de las alternativas de *cálculo de la significación de los puntos del contorno* (véase la sección 4.3.2).
- $n_2 \in \{1, 2, 3, 4, 5, 6\}$ es una de las alternativas de los *métodos de umbralización* (véase la sección 3.3.5).

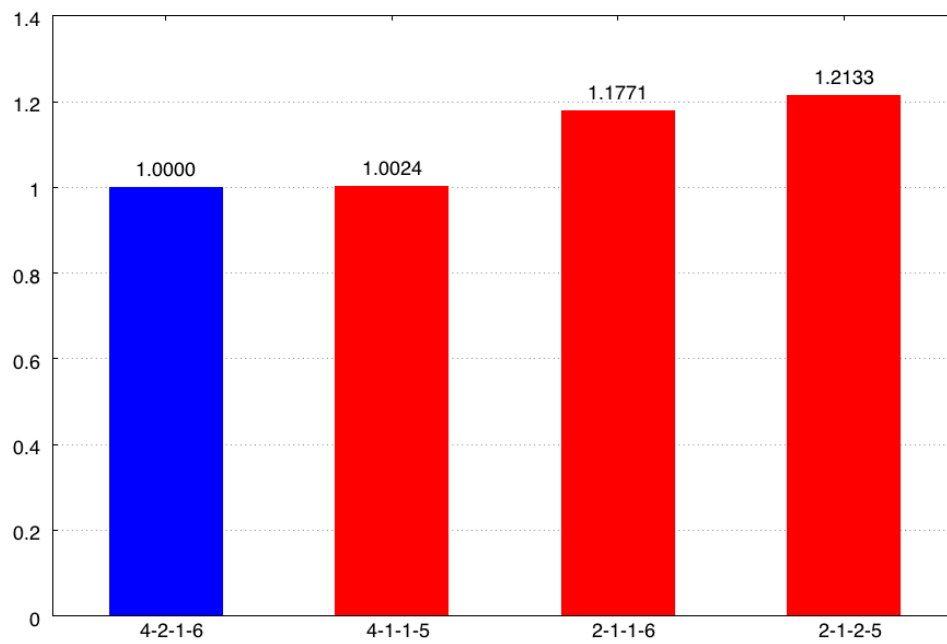
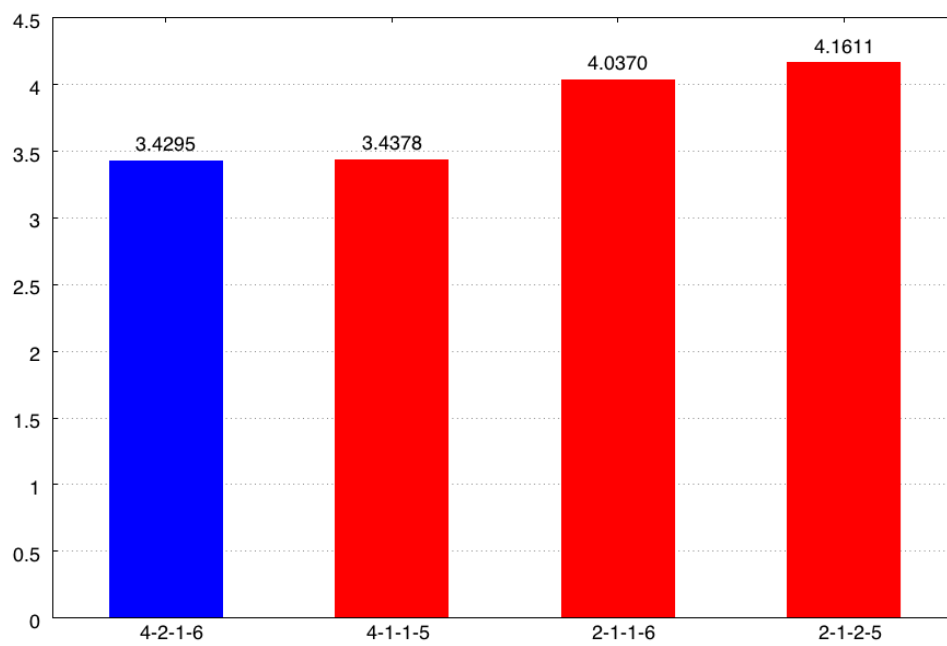
La evaluación cuantitativa de las versiones se muestra en la tabla 4.4 y la figura 4.15, que muestran el valor medio de las medidas de calidad, la desviación típica de la medida WE_2 , la medida combinada $\mu WE_2 + \sigma WE_2$ y el atributo *Index WE₂*. Las figuras 4.16 y 4.17 permiten realizar una evaluación cualitativa de las aproximaciones poligonales generadas por las cuatro mejores alternativas basadas en la estrategia *merge* (*2-5*, *2-6*, *1-5* y *2-2*) para dos contornos de la base de datos de pruebas *MPEG-7*.

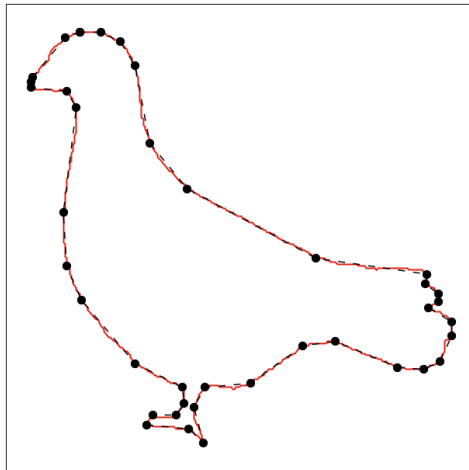
La tabla 4.4 muestra los resultados de las 18 versiones del método, que están ordenadas de forma creciente según el valor de *Index WE₂* (última columna de la tabla). Un análisis más detallado permite realizar las siguientes observaciones:

Tabla 4.2: Primer experimento. Comparativa del rendimiento de las versiones de la estrategia *split*. Las versiones se muestran en orden creciente según el valor de **Index WE₂**.

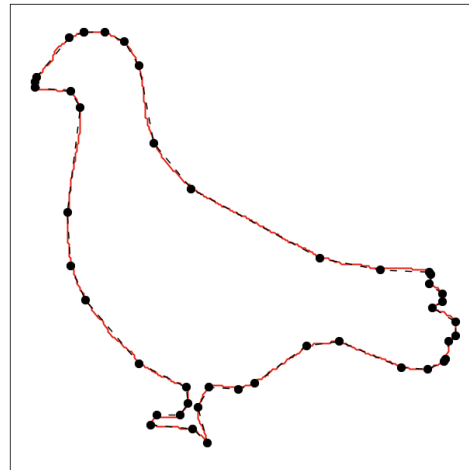
Orden	Versión	μ DP	μ CR	μ E ∞	μ ISE	μ WE ₂	σ WE ₂	μ WE ₂ + σ WE ₂	Index WE ₂
1	4-2-1-6	47,8350	29,2153	2,4477	1749,0967	1,5381	1,8915	3,4295	1,0000
2	4-1-1-6	47,8307	29,2182	2,4480	1750,2157	1,5382	1,8928	3,4309	1,0004
3	4-1-1-5	53,9579	26,4255	2,1024	1396,0293	1,4592	1,9785	3,4378	1,0024
4	4-2-1-5	53,9514	26,4266	2,1024	1396,6875	1,4593	1,9792	3,4384	1,0026
5	4-2-2-5	52,8014	26,9101	2,1666	1419,9355	1,4832	2,0005	3,4837	1,0158
6	4-1-2-5	52,8021	26,9096	2,1666	1419,8584	1,4834	2,0018	3,4852	1,0162
7	4-2-2-6	46,4064	29,9429	2,5589	1846,5844	1,5904	1,9452	3,5356	1,0309
8	4-1-2-6	46,4021	29,9458	2,5592	1847,7034	1,5906	1,9465	3,5370	1,0314
9	4-1-1-2	43,5364	31,7053	2,8059	2242,1162	1,6987	2,0273	3,7260	1,0865
10	4-2-1-2	43,5279	31,7064	2,8063	2243,6757	1,6993	2,0325	3,7318	1,0881
11	4-1-2-2	42,5279	32,3659	2,9305	2336,5634	1,7620	2,0818	3,8438	1,1208
12	4-2-2-2	42,5193	32,3670	2,9310	2338,1228	1,7627	2,0868	3,8494	1,1224
13	2-1-1-6	39,6843	34,6628	3,1999	2506,3510	1,8304	2,2066	4,0370	1,1771
14	2-2-1-6	39,6807	34,6592	3,2013	2507,7682	1,8325	2,2280	4,0605	1,1840
15	3-2-1-6	41,7393	33,0082	3,0064	2363,3855	1,8055	2,2806	4,0861	1,1915
16	3-1-1-6	41,7364	33,0075	3,0067	2363,6160	1,8059	2,2818	4,0877	1,1919
17	2-1-2-6	38,1771	35,7895	3,3360	2648,4523	1,8803	2,2362	4,1165	1,2003
18	3-2-2-6	40,2921	34,1334	3,1279	2486,3785	1,8373	2,2805	4,1178	1,2007
19	3-1-2-6	40,2893	34,1328	3,1282	2486,6090	1,8377	2,2816	4,1194	1,2012
20	1-1-1-6	38,8850	35,1934	3,2596	2607,9931	1,8663	2,2573	4,1236	1,2024
25 (*)	2-1-2-5	45,7986	30,6597	2,7021	1961,6080	1,7115	2,4496	4,1611	1,2133

Nota: (*) la versión 2-1-2-5, en la posición 25 de la tabla, representa al método *split* presentado en el capítulo 3.

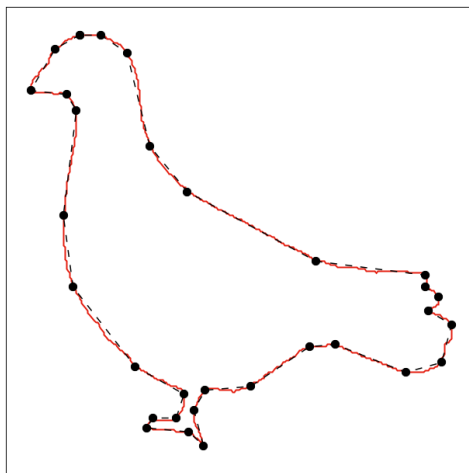
(a) Rendimiento de las versiones *split*. Medida $Index WE_2$ (b) Rendimiento de las versiones *split*. Medida $\mu WE_2 + \sigma WE_2$ **Figura 4.12:** Primer experimento. Gráficas comparativas del rendimiento de las versiones *split*



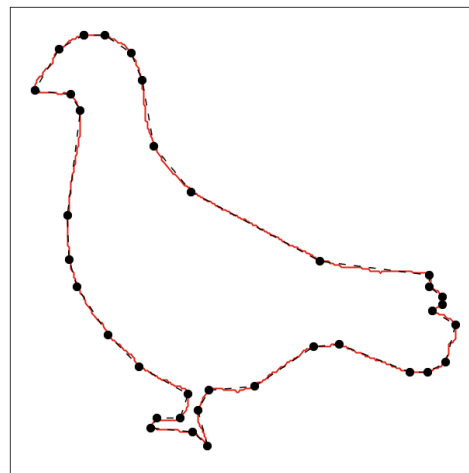
(a) Versión *split* 4-2-1-6
 DP = 39, $WE_2 = 0,8267$



(b) Versión *split* 4-1-1-5
 DP = 44, $WE_2 = 0,9790$



(c) Versión *split* 2-1-1-6
 DP = 31, $WE_2 = 1,1100$



(d) Versión *split* 2-1-2-5
 DP = 36, $WE_2 = 0,8473$

Figura 4.13: Primer experimento: comparación de aproximaciones poligonales del contorno *bird-17* ($N = 779$ puntos) obtenidas por versiones basadas en la estrategia *split*. La versión 2-1-2-5 representa al método presentado en el capítulo 3.

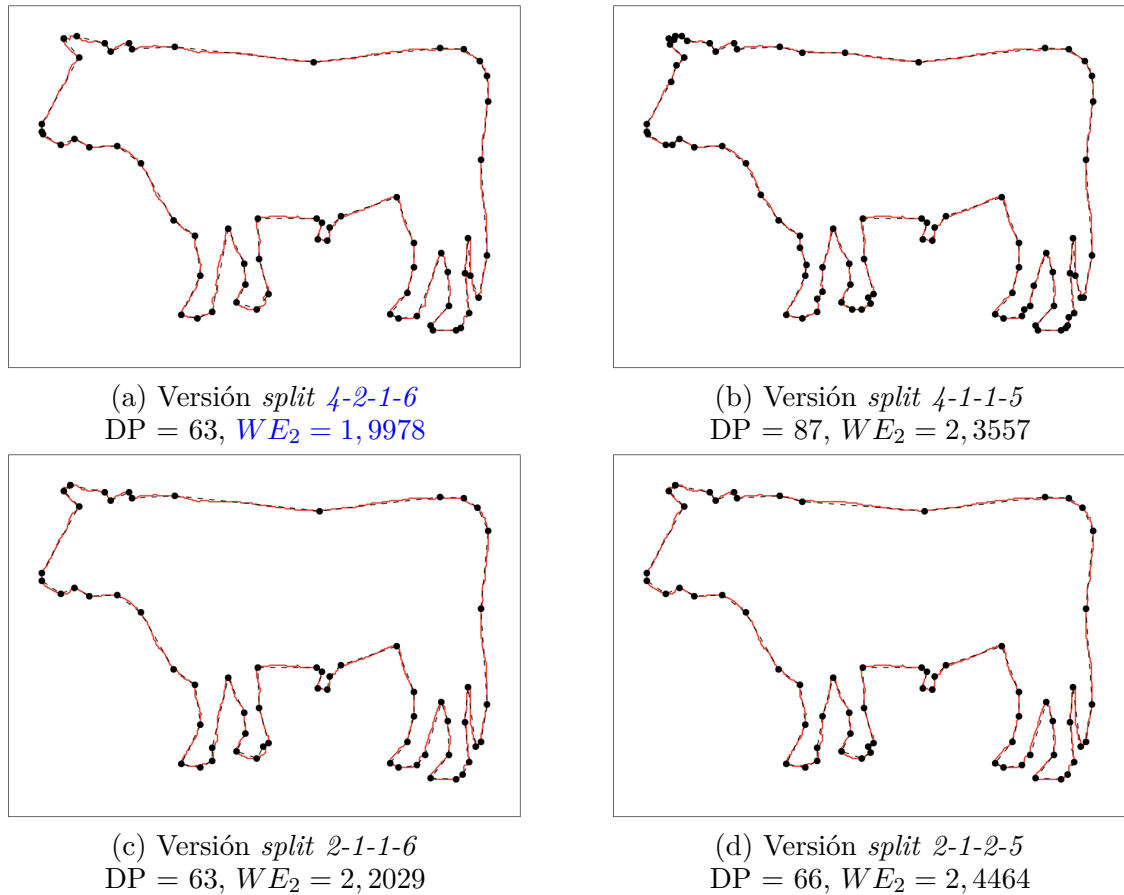


Figura 4.14: Primer experimento: comparación de aproximaciones poligonales del contorno *cattle-6* ($N = 1744$ puntos) obtenidas por versiones basadas en la estrategia *split*. La versión 2-1-2-5 representa al método presentado en el capítulo 3.

Tabla 4.3: Tabla de alternativas del método *merge*

Medida y simetría	Umbralización
1. E_{max} no simétrica	1. Distancia mín. a $(3/N, 0.0)$
2. ISE no simétrica	2. Distancia máx. a $(1.0, 1.0)$
3. Longitud no simétrica	3. Distancia máx. a recta de Rosin
	4. Diagonal principal
	5. Método adaptativo
	6. Método semi-adaptativo

Nota: la combinación de las alternativas permite definir **18 versiones** diferentes.

■ **Posición 1:** versión 2-5

- Esta es la mejor versión basada en la estrategia *merge* y sus alternativas son:
 - **Medida:** 2.- *ISE no simétrica*
 - **Umbralización:** 5.- método adaptativo
- Esta versión es la que ha obtenido el valor medio $\mu WE_2 = 2,6522$ y la mejor desviación típica $\sigma WE_2 = 2,9284$, tomando la medida combinada el valor $\mu WE_2 + \sigma WE_2 = 5,5806$.
- Si se analiza el valor del atributo *Index WE₂*, se comprueba que esta versión supera con claridad a las demás versiones basadas en la estrategia *merge*, dado que sus índices varían desde 1,1157 hasta 3,0424.

■ **Posición 2:** versión 2-6

- Alternativas de esta versión:
 - **Medida:** 2.- *ISE no simétrica*
 - **Umbralización:** 6.- método semi-adaptativo
- Esta versión ha obtenido un valor medio $\mu WE_2 = 3,0037$ y una desviación típica $\sigma WE_2 = 3,2229$. El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 6,2265 y su *Index WE₂* = 1,1157.

■ **Posición 3:** versión 1-5

- Alternativas de esta versión:
 - **Medida:** 1.- *E_{max} no simétrica*
 - **Umbralización:** 5.- método adaptativo
- Esta versión es la que ha obtenido los mejores resultados al analizar el valor medio $\mu WE_2 = 2,4718$ y una desviación típica $\sigma WE_2 = 3,8076$. El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 6,2794 y su *Index WE₂* = 1,1252.

■ **Posición 4:** versión 2-2

- Alternativas de esta versión:
 - **Medida:** 2.- *ISE no simétrica*
 - **Umbralización:** 2.- Distancia máx. a (1.0, 1.0)
- Esta versión ha obtenido un valor medio $\mu WE_2 = 3,2902$, la desviación típica $\sigma WE_2 = 3,5457$. El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 6,8359 y su *Index WE₂* = 1,2249.

Por último, la tabla 4.5 permite comparar los resultados de la mejor versión basada en la estrategia *merge* (versión 2-5) con los resultados de las versiones basadas en la estrategia *split* (4-2-1-6, 4-1-1-5, 2-1-1-6 y 2-1-2-5) que fueron analizadas en el primer experimento (véase la sección 4.4.2). Las figuras 4.18 y 4.19 muestran algunas de las

aproximaciones poligonales generadas por estas versiones para facilitar una evaluación cualitativa.

El análisis de los resultados de la tabla 4.5 permite realizar la siguiente observación:

- La versión *2-5* basada en la estrategia *merge* es la que ha obtenido los mejores resultados. Sin embargo, no consigue superar a las versiones basadas en la estrategia *split* ni a la versión correspondiente al método original propuesto en el capítulo 3. En concreto, la versión *2-5* del método *merge* ocupa la posición número 73 del ranking si se la compara con todas las demás versiones de la estrategia *split*, con un valor $Index WE_2$ de 1,6272 (véase la tabla 4.5).

En resumen, este segundo experimento demuestra que la estrategia *merge* no permite definir una versión que supere el rendimiento de las mejores versiones *split* evaluadas en el experimento anterior (versiones *4-2-1-6* y *4-1-1-6*) (véase la tabla 4.5).

4.5. Conclusiones

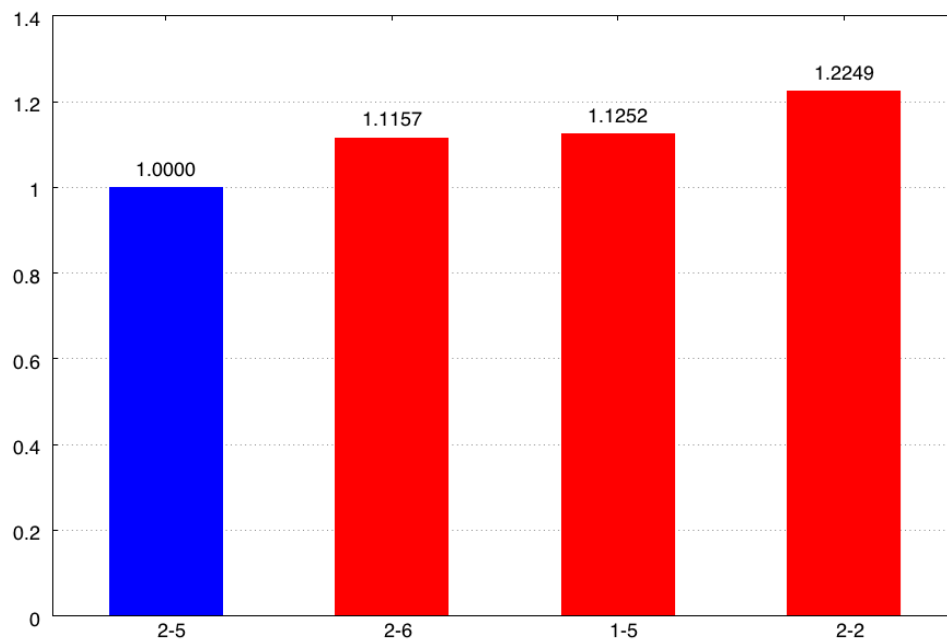
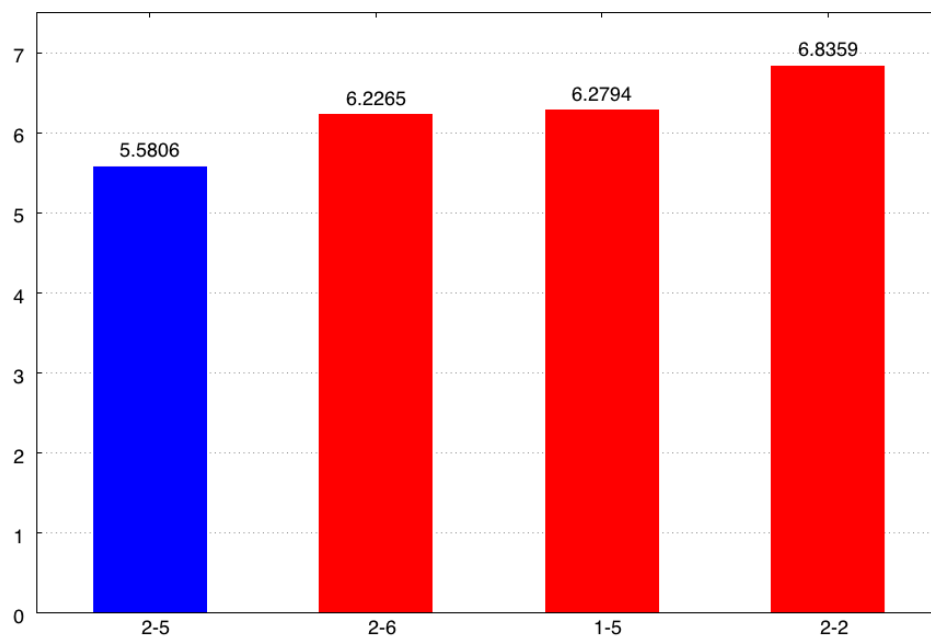
En este capítulo se ha realizado un estudio exhaustivo del método original propuesto en el capítulo 3. Para ello se han establecido dos estrategias. La primera estrategia está basada en la idea de división (*split*) del contorno y consta de siete etapas, siendo la última opcional. Cuatro de estas etapas poseen alternativas que permiten definir 288 versiones diferentes del método original. La segunda estrategia está basada en la idea de fusión (*merge*) de puntos del contorno y consta de seis etapas. Dos de estas etapas poseen alternativas que permiten definir 18 versiones diferentes del método original. Las nuevas versiones no requieren de ningún parámetro, siendo, por tanto, versiones automáticas. La complejidad computacional de las versiones (*split*) o (*merge*) definidas es $\mathcal{O}(n \log(n))$.

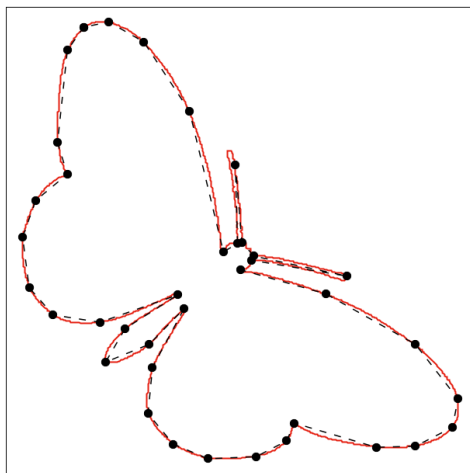
De los experimentos realizados para evaluar el rendimiento de estas versiones del método, se concluye que:

- Hay 24 versiones basadas en la estrategia *split* que han obtenido mejores resultados que la versión *2-1-2-5*, que se corresponde con el método original (propuesto en el capítulo 3).
- La versión *4-2-1-6*, basada en el uso del *casco convexo*, es la que ha obtenido los mejores resultados entre este tipo de versiones. Sin embargo, esta versión requiere la aplicación de una fase final de supresión de puntos *superfluos* que están *casi alineados*. La versión *4-1-1-6* obtiene resultados similares a la versión *4-2-1-6*.
- De las versiones basadas en la estrategia *merge*, la versión *2-5* es la que ha obtenido los mejores resultados, pero no supera a los obtenidos por el método original (*2-1-2-5*) ni a las mejores versiones basadas en la estrategia *split* (*4-2-1-6* y *4-1-1-6*).

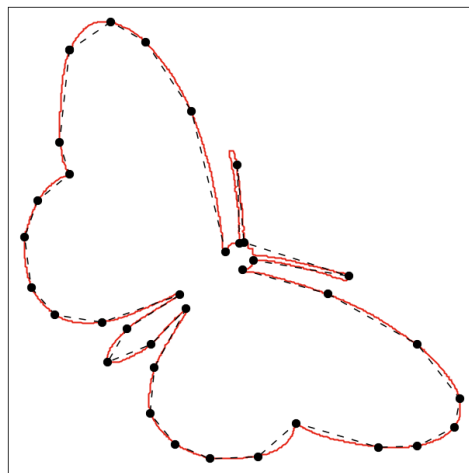
Tabla 4.4: Segundo experimento. Comparativa del rendimiento de las alternativas de la estrategia *merge*. Las versiones se muestran en orden creciente según el valor de **Index WE₂**.

Orden	Versión	μ DP	μ CR	μ E ∞	μ ISE	μ WE ₂	σ WE ₂	μ WE ₂ + σ WE ₂	Index WE ₂
1	2-5	26,1271	50,5515	6,3057	9236,7341	2,6522	2,9284	5,5806	1,0000
2	2-6	23,2814	56,0000	7,4279	12563,6366	3,0037	3,2229	6,2265	1,1157
3	1-5	43,0950	27,5188	3,4057	2811,2151	2,4718	3,8076	6,2794	1,1252
4	2-2	21,6236	59,9894	8,2685	16014,0010	3,2902	3,5457	6,8359	1,2249
5	1-6	33,8986	34,8433	4,4807	4556,1293	2,9201	4,0713	6,9915	1,2528
6	1-2	29,5907	40,1021	5,3523	6443,8153	3,2951	4,3405	7,6355	1,3682
7	1-1	28,7300	41,1158	5,6840	7266,7074	3,4527	4,4725	7,9252	1,4201
8	2-1	19,5607	65,4405	9,7985	22811,7424	3,8791	4,1828	8,0619	1,4446
9	2-3	18,2543	70,7298	10,7393	26785,8193	4,0110	4,2078	8,2187	1,4727
10	1-4	27,1371	43,5057	6,3756	8949,6306	3,7838	4,9122	8,6960	1,5583
11	2-4	17,5300	72,3227	11,8896	33214,9201	4,5182	4,7603	9,2785	1,6626
12	1-3	21,6243	57,0679	9,3025	26560,0710	4,8722	6,2362	11,1084	1,9905
13	3-5	37,2464	38,8720	6,0691	9575,6945	5,6249	7,4601	13,0850	2,3447
14	3-6	31,6514	44,9230	7,1798	13568,2733	6,1219	7,7646	13,8865	2,4883
15	3-2	28,6586	49,4161	7,9500	17120,7175	6,5016	9,2173	15,7189	2,8167
16	3-1	27,2029	51,5670	8,5569	19376,7498	6,9166	9,1099	16,0266	2,8718
17	3-3	22,7121	63,0797	10,6414	32346,2159	7,4017	9,3149	16,7166	2,9955
18	3-4	25,6964	55,2423	9,3743	22712,4637	7,4019	9,5769	16,9787	3,0424

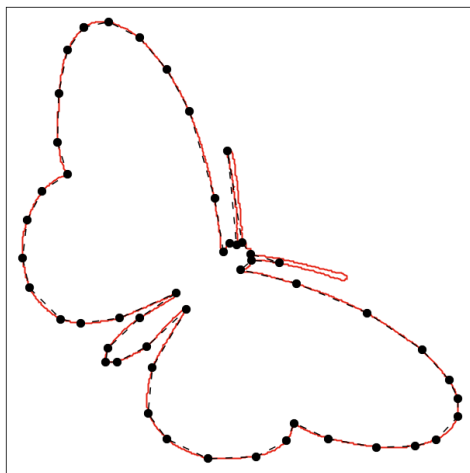
(a) Rendimiento de las versiones *merge*. Medida $Index WE_2$ (b) Rendimiento de las versiones *merge*. Medida $\mu WE_2 + \sigma WE_2$ **Figura 4.15:** Segundo experimento. Gráficas comparativas del rendimiento de las versiones *merge*



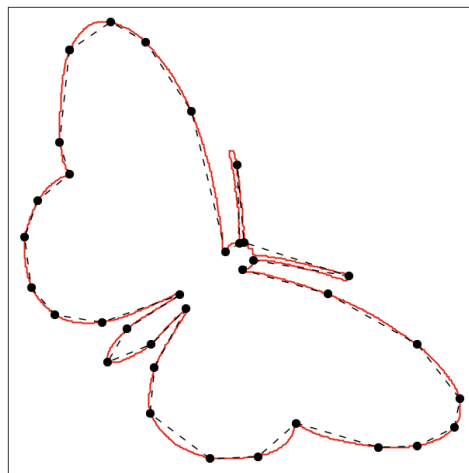
(a) Versión *merge 2-5*
DP = 38, $WE_2 = 3,3255$



(b) Versión *merge 2-6*
DP = 35, $WE_2 = 4,2292$



(c) Versión *merge 1-5*
DP = 50, $WE_2 = 3,5780$



(d) Versión *merge 2-2*
DP = 34, $WE_2 = 4,6841$

Figura 4.16: Segundo experimento: comparación de aproximaciones poligonales del contorno *butterfly-20* ($N = 1865$ puntos) obtenidas por versiones basadas en la estrategia *merge*.

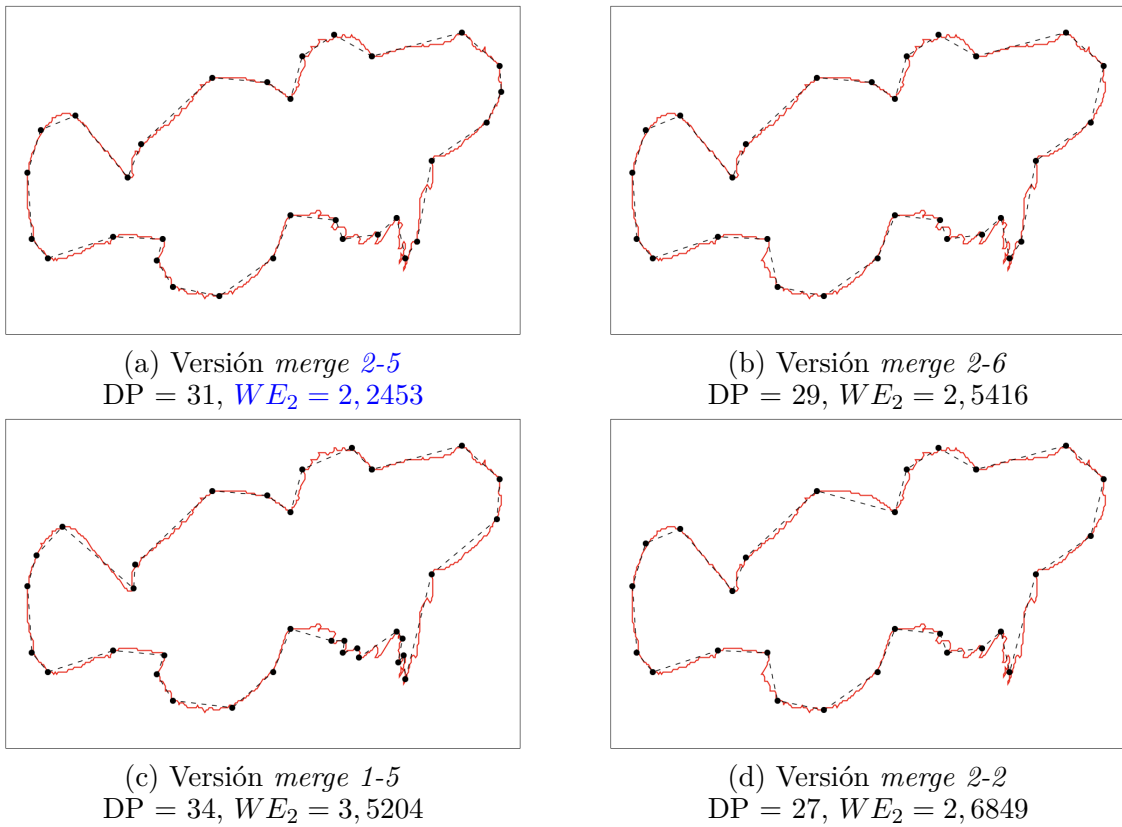


Figura 4.17: Segundo experimento: comparación de aproximaciones poligonales del contorno *fish-7* ($N = 866$ puntos) obtenidas por versiones basadas en la estrategia *merge*.

Tabla 4.5: Comparativa del rendimiento de las mejores versiones *split* y *merge*. Las versiones se muestran en orden creciente según el valor de **Index WE₂**.

Estrategia	Versión	μ DP	μ CR	μ E _{max}	μ ISE	μ WE ₂	σ WE ₂	μ WE ₂ + σ WE ₂	Index WE ₂
Split (1)	4-2-1-6	47,8350	29,2153	2,4477	1749,0967	1,5381	1,8915	3,4295	1,0000
Split (3)	4-1-1-5	53,9579	26,4255	2,1024	1396,0293	1,4592	1,9785	3,4378	1,0024
Split (13)	2-1-1-6	39,6843	34,6628	3,1999	2506,3510	1,8304	2,2066	4,0370	1,1771
Split (25) (*)	2-1-2-5	45,0692	30,9705	2,7576	2024,6729	1,7507	2,5393	4,2901	1,2133
Merge (1)	2-5	26,1271	50,5515	6,3057	9236,7341	2,6522	2,9284	5,5806	1,6272

Notas:

Entre paréntesis se muestra la posición de cada versión en las tablas originales (4.2 y 4.4)

(*) La versión *2-1-2-5* representa al método *split* presentado en el capítulo 3.

El valor de **Index WE₂** ha sido recalculado para las versiones que se muestran en la tabla.

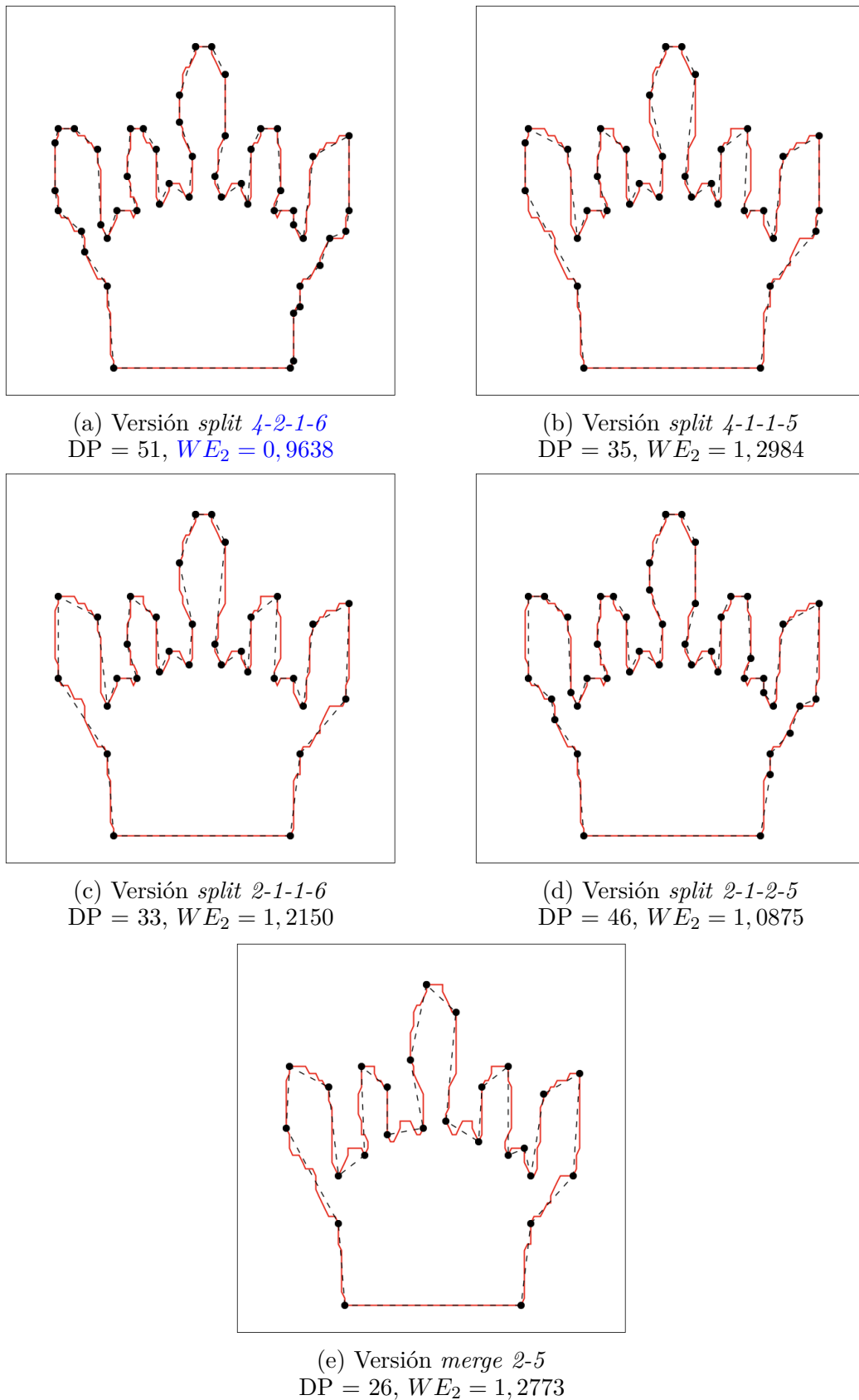
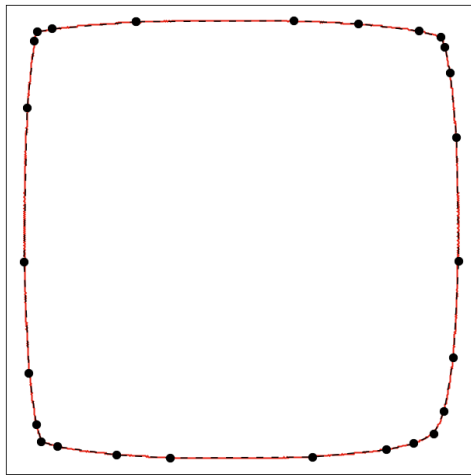
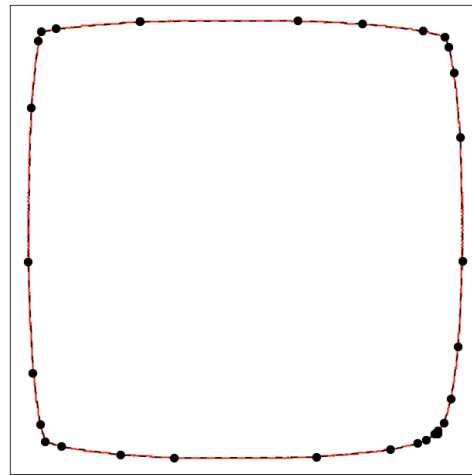


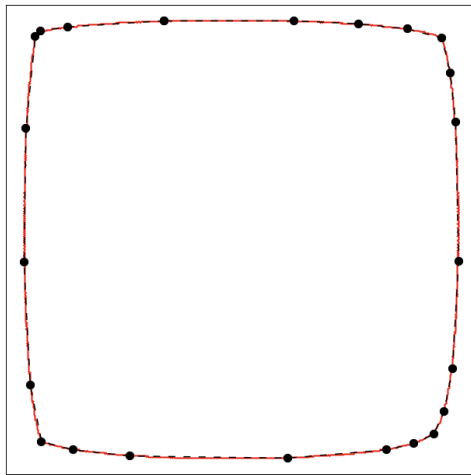
Figura 4.18: Comparación de aproximaciones poligonales del contorno del contorno *crown-12* ($N = 335$ puntos) obtenidas por versiones *split* y *merge*. La versión 2-1-2-5 representa al método *split* presentado en el capítulo 3.



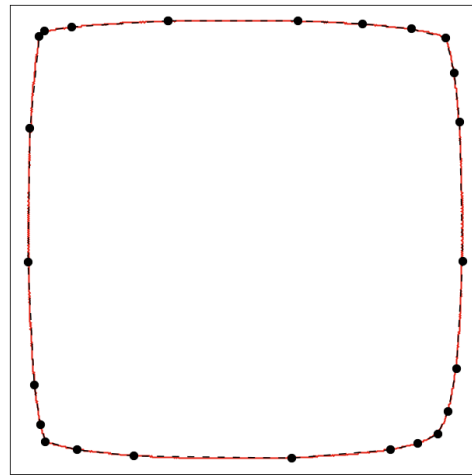
(a) Versión *split* 4-2-1-6
 DP = 26, $WE_2 = 0,0935$



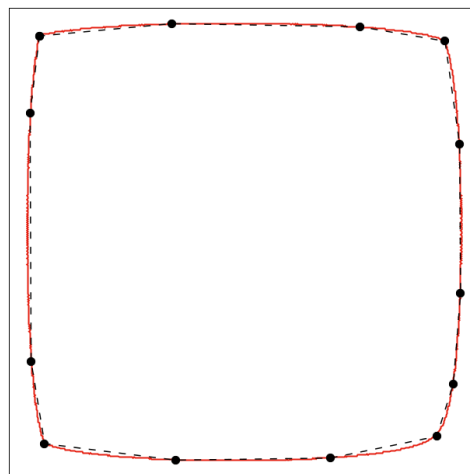
(b) Versión *split* 4-1-1-5
 DP = 30, $WE_2 = 0,1186$



(c) Versión *split* 2-1-1-6
 DP = 23, $WE_2 = 0,1175$



(d) Versión *split* 2-1-2-5
 DP = 24, $WE_2 = 0,1162$



(e) Versión *merge* 2-5
 DP = 13, $WE_2 = 0,4069$

Figura 4.19: Comparación de aproximaciones poligonales del contorno del contorno *device3-7* ($N = 1547$ puntos) obtenidas por versiones *split* y *merge*. La versión 2-1-2-5 representa al método *split* presentado en el capítulo 3.

5

Optimización de los métodos basados en las estrategias *split* y *merge*

5.1. Introducción

En el capítulo 4 se ha realizado un análisis exhaustivo de diferentes extensiones del método original propuesto en el capítulo 3. Dicho estudio ha utilizado las estrategias *split* y *merge* para definir nuevas versiones. En este capítulo, se abordará una posible mejora de dichas versiones mediante la incorporación de una etapa final de optimización inspirada en el método propuesto por *Masood* [57].

Los métodos *split* y *merge* optimizados mantienen todas las etapas originales descritas en las secciones 4.2 y 4.3, pero incorporan una etapa final en la que se implementa un método de optimización basado en el método propuesto por *Masood*.

En las figuras 5.1 y 5.2 se muestran los métodos *split* y *merge* optimizados. Nótese que la etapa de optimización, que es la variación introducida respecto de las versiones propuestas en el capítulo anterior, se resalta en color amarillo.

La organización del capítulo es la siguiente: la sección 5.2 explica el método de optimización; los experimentos realizados y los resultados obtenidos se describen en la sección 5.3 y, finalmente, la sección 5.4 resume las principales conclusiones obtenidas.

5.2. Método de optimización

Se propone un método de optimización inspirado en el método de *Masood*, que permite mejorar el ajuste de una aproximación poligonal a partir del desplazamiento de sus puntos dominantes. Estos desplazamientos reducen el error de la aproximación poligonal respecto del contorno de entrada (véase la figura 5.3). Es preciso resaltar que el proceso *no incrementa el número de puntos* de la aproximación, sino que los desplaza a una nueva posición en la que contribuyen a la obtención de un mejor ajuste global de la figura. El proceso es el siguiente:

1. Inicialmente todos los puntos dominantes del contorno son considerados *puntos inestables*.
 - Sea $L = \{P_i, \dots, P_n\}$ el conjunto de puntos inestables del contorno.
2. Mientras que L tenga puntos inestables, para cada punto inestable $P_i \in L$:
 - a) Obtener los puntos *anterior* $P_{i-1} \in L$ y *posterior* $P_{i+1} \in L$.
 - b) Calcular la posición *óptima* del punto P_i :
 - 1) Calcular la recta $r \equiv aX + bY + c = 0$ que pasa por P_{i-1} y P_{i+1} .
 - 2) Para cada punto del contorno P situado entre los puntos $P_{i-1} \in L$ y $P_{i+1} \in L$, calcular el valor *ISE* según la expresión 4.4.
 - 3) Encontrar el punto P_{min} que minimiza el valor *ISE*.
 - c) Si el punto P_i es igual al punto con posición óptima P_{min} , entonces:
 - 1) Marcar el punto P_i como estable.
 - d) Si el punto P_i no es igual al punto P_{min} , entonces:
 - 1) Cambiar el punto P_i por el punto P_{min} . Marcar el punto P_{min} como estable.
 - 2) Marcar como estables los puntos anterior y posterior a P_{min} , es decir, P_{i-1} y P_{i+1} , respectivamente.

Al inicio del método el conjunto L contiene el conjunto de puntos dominantes del contorno, que son marcados como *inestables*. A medida que avanza el proceso, se estabilizarán los puntos. Al finalizar, L contendrá los puntos dominantes de la aproximación optimizada.

En la figura 5.4 se muestran los resultados de aplicar el proceso de optimización a las versiones *split* (2-1-2-5 y 4-1-1-5) usando el contorno *chicken-10_0* (N = 468 puntos). En estas figuras se aprecia que el método de optimización mejora de forma notable el ajuste de la aproximación poligonal. Esta mejora se debe a que el error de ajuste se reduce al desplazar los puntos dominantes. Nótese que este método, como ya se ha mencionado, no incrementa el número de puntos totales del contorno, sino que los distribuye de forma que se minimiza el ajuste de la aproximación poligonal al contorno.

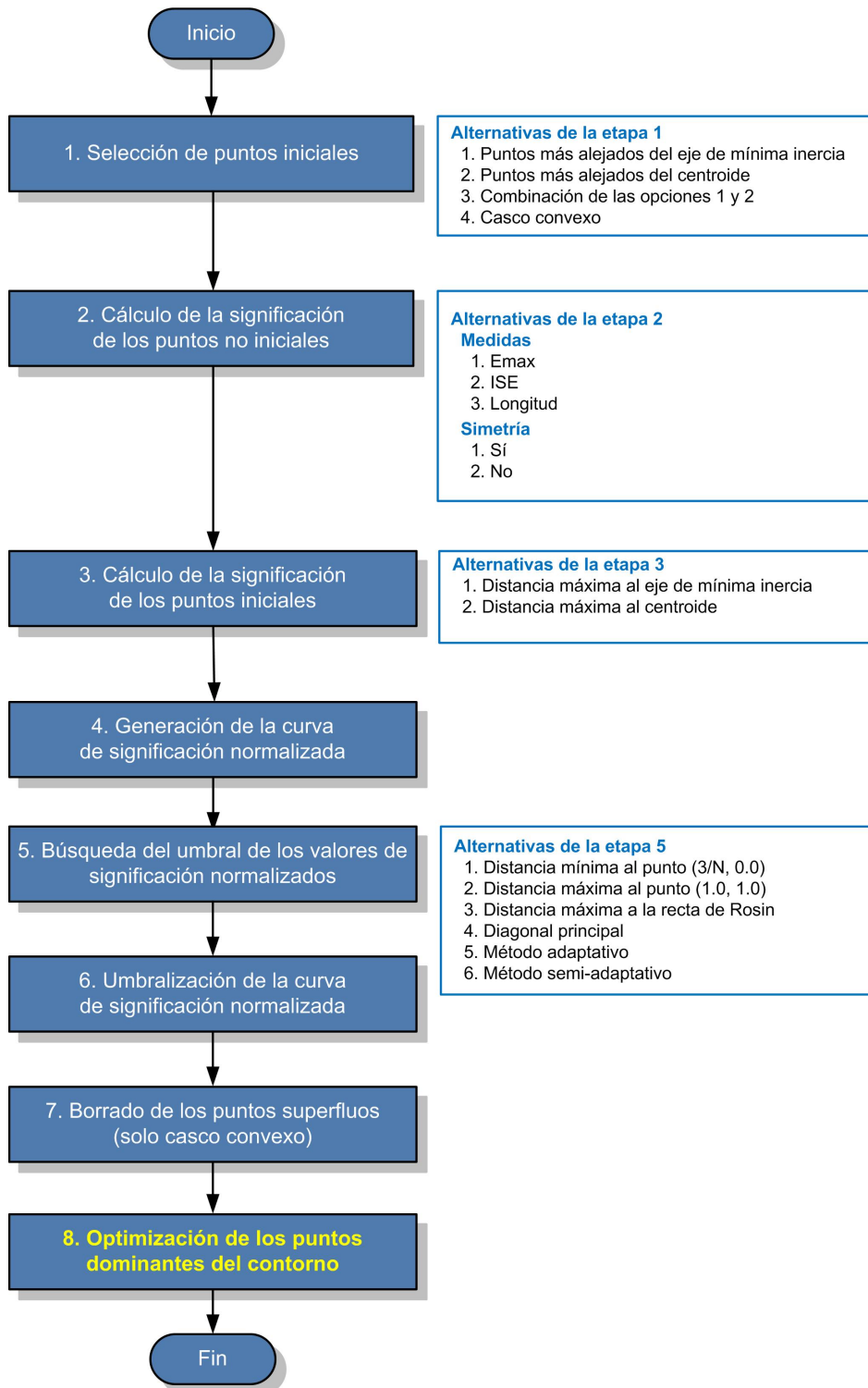


Figura 5.1: Esquema del método optimizado de la estrategia *split*. Se resalta en color amarillo la etapa final de optimización.

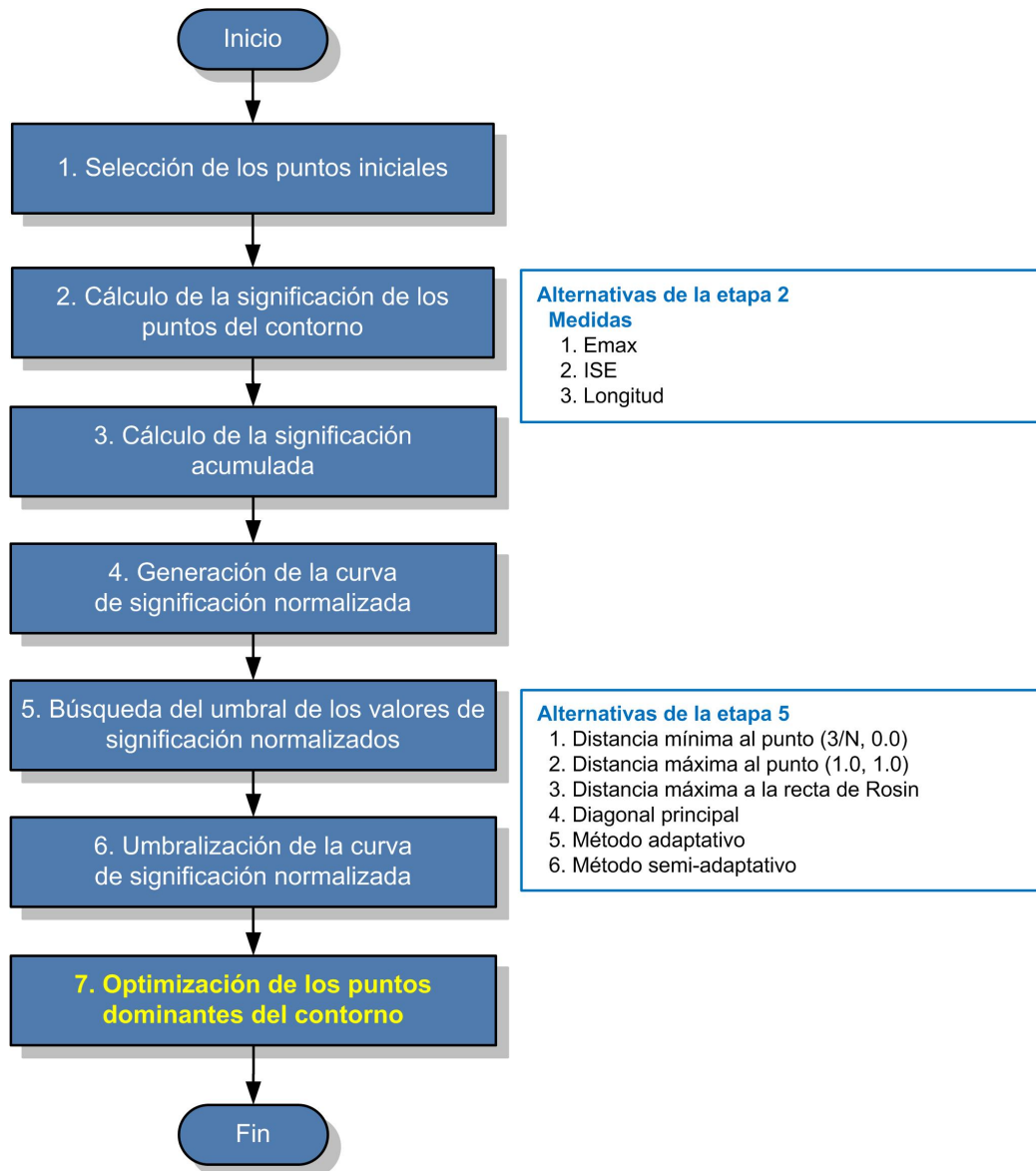
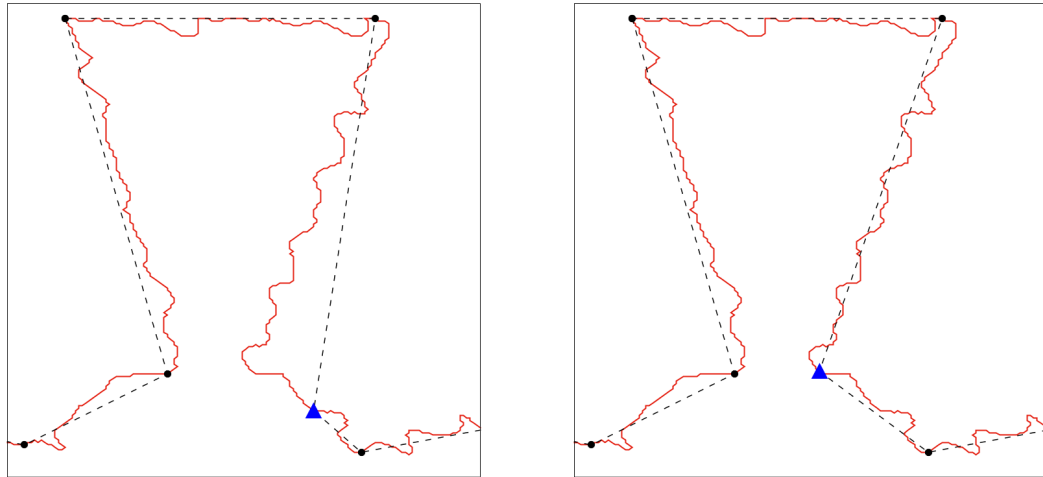


Figura 5.2: Esquema del método optimizado de la estrategia *merge*. Se resalta en color amarillo la etapa final de optimización.

En conclusión, este método permite modificar la ubicación de cada punto P_i entre sus vecinos (P_{i-1} y P_{i+1}), lo que contribuye a una reducción del error general (*ISE*) de la aproximación poligonal resultante (véase la figura 5.3). La complejidad computacional de esta etapa es $\mathcal{O}(d \cdot n)$, donde d es el número de puntos dominantes que deben ser optimizados y n el número de puntos del contorno, respectivamente. Por lo general, el valor d suele ser muy pequeño en comparación con el número de puntos del contorno n , siendo la complejidad, en estos casos, de tipo lineal $\mathcal{O}(n)$.



(a) Aproximación poligonal original

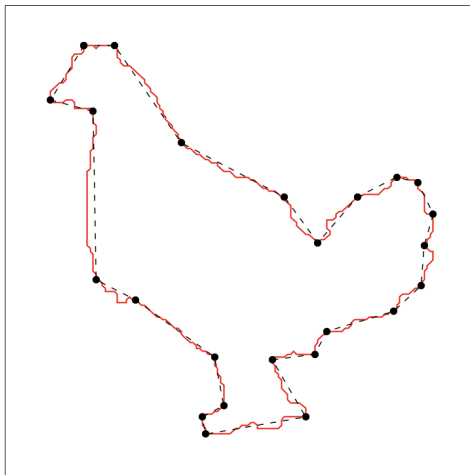
(b) Aproximación optimizada
(se ha optimizado el punto azul)

Figura 5.3: *Proceso de optimización de un punto dominante en el contorno device-5_18.* El punto que se ha optimizado se ha resaltado en azul y se ha marcado con una forma triangular. (a) Aproximación original de la figura; (b) El punto azul se desplaza a su posición *óptima*, con lo que se reduce el error de la aproximación poligonal resultante.

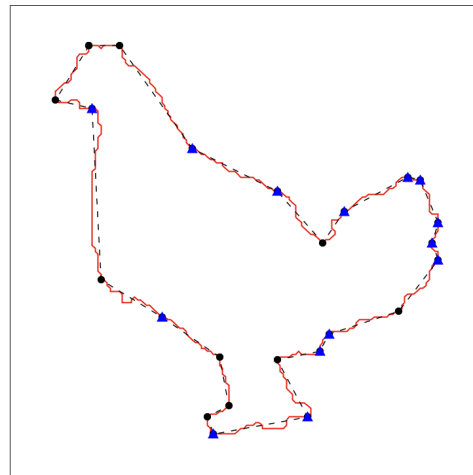
5.3. Evaluación de los métodos y análisis de los resultados

Para evaluar y comparar las versiones optimizadas de las estrategias *split* y *merge*, se han desarrollado dos experimentos:

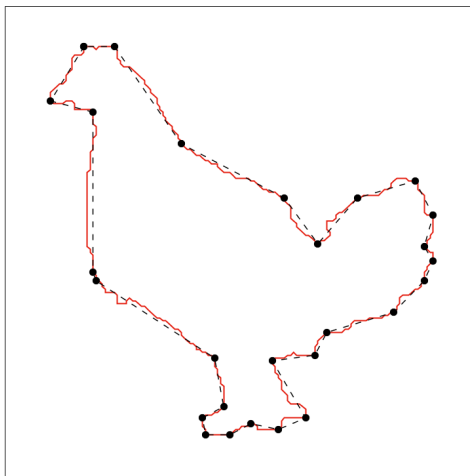
- Primer experimento: selección de la mejor versión del método basado en la estrategia *split* optimizada.
 - Se han redefinido las 288 versiones *split*, propuestas en el capítulo 4, y se ha incorporado la etapa adicional de optimización. Este experimento ha evaluado el rendimiento de las 288 versiones para seleccionar la mejor versión *split* optimizada.



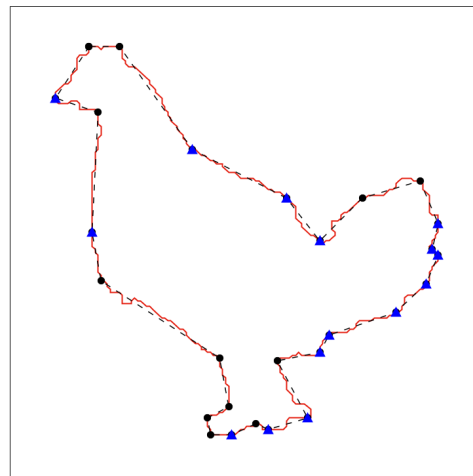
(a) Versión *split* 2-1-2-5
 DP = 27, $WE_2 = 1,7421$



(b) Versión *split* 2-1-2-5 optimizada
 DP = 27, $WE_2 = 1,0452$



(c) Versión *split* 4-1-1-5
 DP = 24, $WE_2 = 1,8074$



(d) Versión *split* 4-1-1-5 optimizada
 DP = 24, $WE_2 = 1,1691$

Figura 5.4: Aplicación del proceso de optimización en las versiones *split* con el contorno *chicken-10_0* ($N = 468$ puntos): (a) versión 2-1-2-5; (b) versión 2-1-2-5 optimizada; (c) versión 4-1-1-5; (d) versión 4-1-1-5 optimizada. Los puntos que se han modificado se han resaltado con un triángulo azul.

- Este primer experimento se puede consultar en la sección 5.3.1.
- Segundo experimento: selección de la mejor versión del método basado en la estrategia *merge* optimizada.
 - Se han redefinido las 18 versiones *merge*, propuestas en el capítulo 4, y se ha añadido la etapa adicional de optimización. Este experimento ha evaluado el rendimiento de las 18 versiones para seleccionar la mejor versión *merge* optimizada.
 - El segundo experimento se puede consultar en la sección 5.3.2.

Al igual que en los capítulos 3 y 4, se ha empleado la misma base de datos *MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)* [42, 87, 92].

Para evaluar la calidad de las aproximaciones poligonales de cada experimento se han utilizado las siguientes medidas: número de puntos dominantes (*DP*), ratio de compresión (*CR*), error cuadrático integral (*ISE*), error máximo (E_∞) y la suma ponderada del error cuadrático (WE_2) (véase la sección 2.2). También se ha utilizado la medida combinada $\mu WE_2 + \sigma WE_2$ para evaluar la estabilidad del rendimiento de cada una de las versiones de los métodos. Por último, se ha calculado la medida de comparación *Index WE₂* (véanse las tablas 4.2 y 4.4).

5.3.1. Primer experimento: selección de la mejor versión optimizada del método *split*

En este experimento se ha añadido el sufijo $-O$ a la notación de las versiones para distinguir las de las versiones *split* sin optimizar que se presentaron en el capítulo anterior.

Por ejemplo, el identificador 2-1-2-5-O representa a la versión que utiliza:

- La alternativa nº 2 para seleccionar los puntos iniciales, es decir, la máxima distancia al centroide.
- La alternativa nº 1 para utilizar la medida *E_{max}* y la simetría al calcular la significación de los puntos no iniciales.
- La alternativa nº 2 para usar la máxima distancia al centroide al calcular la significación de los puntos iniciales.
- La alternativa nº 5 para aplicar el método de umbralización adaptativo.
- Aplica la etapa final de optimización según el método propuesto en la sección 5.2

La evaluación cuantitativa de las versiones se muestra en la tabla 5.1 y en la figura 5.5, que muestran el valor medio de las medidas de calidad, la desviación típica de la

medida WE_2 , la medida combinada $\mu WE_2 + \sigma WE_2$ y el atributo $Index WE_2$. La tabla 5.1 muestra los resultados de las 20 mejores versiones del método optimizado. Esta tabla se encuentra ordenada de forma creciente según el valor de $Index WE_2$ (última columna de la tabla). Un análisis más detallado permite realizar las siguientes observaciones:

■ **Posición 1:** versión *4-2-1-6-O*

- Las alternativas de esta versión son las siguientes:
 - **Selección de iniciales:** 4.- Casco convexo
 - **Medida y simetría:** 2.- Emax *no simétrica*
 - **Significación de iniciales:** 1.- Distancia máxima al centroide
 - **Umbralización:** 6.- Método semi-adaptativo
 - **Optimización:** Método de optimización.
- Esta versión se mantiene en la primera posición porque ha obtenido el mejor valor combinado $\mu WE_2 + \sigma WE_2 = 1,9567$, siendo, por tanto, la versión más estable entre las 288 versiones analizadas. Si se analizan de forma independiente la media y la desviación típica, esta versión obtiene el quinto mejor valor para la media $\mu WE_2 = 0,8928$ y el mejor valor para la desviación típica $\sigma WE_2 = 1,0638$.
- Esta versión utiliza la variante de selección de puntos iniciales basada en el *casco convexo*, por lo que presenta el inconveniente de tener que aplicar la séptima etapa de supresión de puntos superfluos (véanse la sección 4.2.4 y la figura 4.8).
- Esta variante se mantiene en la primera posición del ranking. Además, la variante mejora respecto de la versión 4-2-1-6 original, puesto que el valor de la medida combinada $\mu WE_2 + \sigma WE_2$ de la versión split 4-2-1-6 es de 3,4295 y la versión split 4-2-1-6-O optimizada posee un valor de 1,9567 (véanse las tablas 4.2 y 5.1).

■ **Posición 6:** versión *4-1-1-5-O*

- Las alternativas de esta versión son las siguientes:
 - **Selección de iniciales:** 4.- Casco convexo
 - **Medida y simetría:** 1.- Emax *simétrica*
 - **Significación de iniciales:** 1.- Distancia máxima al centroide
 - **Umbralización:** 5.- Método adaptativo
 - **Optimización:** Método de optimización.
- Esta versión desciende tres posiciones respecto a la clasificación de las versiones sin optimización de la tabla 4.2.
- Entre las 288 versiones analizadas, esta versión es la que ha obtenido el mejor valor promedio $\mu WE_2 = 0,8710$; sin embargo, su desviación típica es el séptimo valor más pequeño ($\sigma WE_2 = 1,1251$), lo que provoca que la medida combinada $\mu WE_2 + \sigma WE_2 = 1,9961$ coloque a esta versión en la cuarta posición de la tabla, siendo su $Index WE_2 = 1,0201$.

- Esta versión también utiliza la variante de selección de puntos iniciales basada en el *casco convexo*, por lo que presenta el inconveniente de tener que suprimir los puntos superfluos, como ya se ha indicado en la versión anterior.
 - Esta variante ha mejorado de forma sustancial con respecto a su versión original, puesto que el valor de la medida combinada $\mu W E_2 + \sigma W E_2$ de la versión *split* 4-1-1-5 es de 3, 4377 y la versión *split* 4-1-1-5-O optimizada posee un valor de 1, 9961 (véanse las tablas 4.2 y 5.1).
- **Posición 14:** versión *2-1-2-5-O*
- Esta versión mejora su posición debido a la optimización y asciende desde la posición 25 de la tabla 4.2 hasta la posición 14.
 - Esta es la versión que optimiza el método propuesto en la sección 3.3. Sus alternativas son las siguientes:
 - **Selección de iniciales:** 2.- Máxima distancia al centroide.
 - **Medida y simetría:** 1.- Emax *simétrica*
 - **Significación de iniciales:** 2.- Distancia máxima al centroide
 - **Umbralización:** 5.- Método adaptativo
 - **Optimización:** Método de optimización.
 - El valor obtenido para la medida combinada $\mu W E_2 + \sigma W E_2$ es 2,1035 y su $Index W E_2 = 1, 0750$.
 - Esta variante mejora respecto a su versión original, puesto que el valor de la medida combinada $\mu W E_2 + \sigma W E_2$ de la versión *split* 2-1-2-5 es de 4, 1611 y la versión *split* 2-1-2-5-O optimizada posee un valor de 2, 1035 (véanse las tablas 4.2 y 5.1).
- **Posición 17:** versión *2-1-1-6-O*
- Las alternativas de esta versión son las siguientes:
 - **Selección de iniciales:** 2.- Distancia al centroide
 - **Medida y simetría:** 1.- Emax *simétrica*
 - **Significación de iniciales:** 1.- Distancia máxima al centroide
 - **Umbralización:** 6.- Método semi-adaptativo
 - **Optimización:** Método de optimización.
 - Esta versión empeora su posición respecto a la clasificación de las versiones sin optimización de la tabla 4.2. El valor obtenido para la medida combinada $\mu W E_2 + \sigma W E_2$ es 2,1183 y su $Index W E_2 = 1, 0826$.
 - Esta variante mejora respecto a su versión original, puesto que el valor de la medida combinada $\mu W E_2 + \sigma W E_2$ de la versión *split* 2-1-1-6 es de 4, 0605 y la versión *split* 2-1-1-6-O optimizada posee un valor de 2, 1183 (véanse las tablas 4.2 y 5.1).

Las figuras 5.6 y 5.7 permiten realizar una evaluación cualitativa de las aproximaciones poligonales generadas por las cuatro alternativas estudiadas del método *split* (*4-2-1-6-O*, *4-1-1-5-O*, *2-1-2-5-O* y *2-1-1-6-O*) para dos contornos de la base de datos de pruebas *MPEG-7*. Además, en las figuras 5.8 y 5.9 se muestra una comparativa de las versiones *4-2-1-6-O*, *4-1-1-5-O*, *2-1-2-5-O* y *2-1-1-6-O* respecto a sus versiones originales (sin optimizar).

Un análisis de los resultados de este primer experimento permite extraer las siguientes conclusiones:

- En la tabla 5.1 se indica que el proceso de optimización de la estrategia *split* ha permitido obtener 13 versiones (de las 288 posibles) que tienen un mejor rendimiento que el método original.
- La versión *4-2-1-6-O*, basada en el casco convexo para seleccionar los puntos iniciales, es la que ha obtenido los mejores resultados si se tiene en cuenta la medida combinada $\mu WE_2 + \sigma WE_2$. Esta versión requiere de una fase final para suprimir puntos superfluos casi alineados (véase la sección 4.2.4).

Al igual que en el capítulo 4, la versión *4-2-1-6-O* es la que obtiene los mejores resultados, pero la versión *4-1-1-6-O* también obtiene resultados muy similares en la medida combinada $\mu WE_2 + \sigma WE_2$. Esto implica que la versión candidata *split* estaría compuesta por las siguientes alternativas:

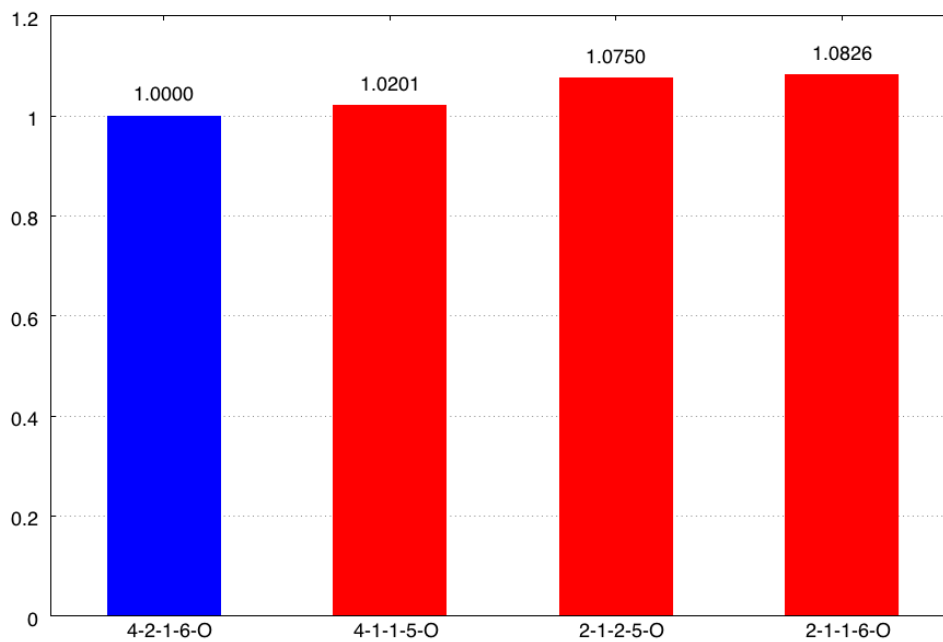
- **Selección de iniciales:** 4.- Casco convexo
- **Medida y simetría:** 1.- Emax *simétrica* o 2.- Emax *no simétrica*
- **Significación de iniciales:** 1.- Distancia máxima al centroide
- **Umbralización:** 6.- Método semi-adaptativo
- **Optimización:** Método de optimización.

Las versiones candidatas (*4-2-1-6-O* y *4-1-1-6-O*) hacen uso del casco convexo como método de selección de puntos iniciales y utilizan el mecanismo de significación de puntos iniciales de la distancia máxima al centroide. En la etapa de umbralización, el método que ahora obtiene mejores resultados es el sexto método propuesto, el método semi-adaptativo. En cambio, en la etapa de medida y simetría se tienen dos posibilidades. La versión resultante arrojará resultados similares y sin diferencias significativas tanto si se emplea la medida Emax *simétrica* como si se emplea la medida Emax *no simétrica*.

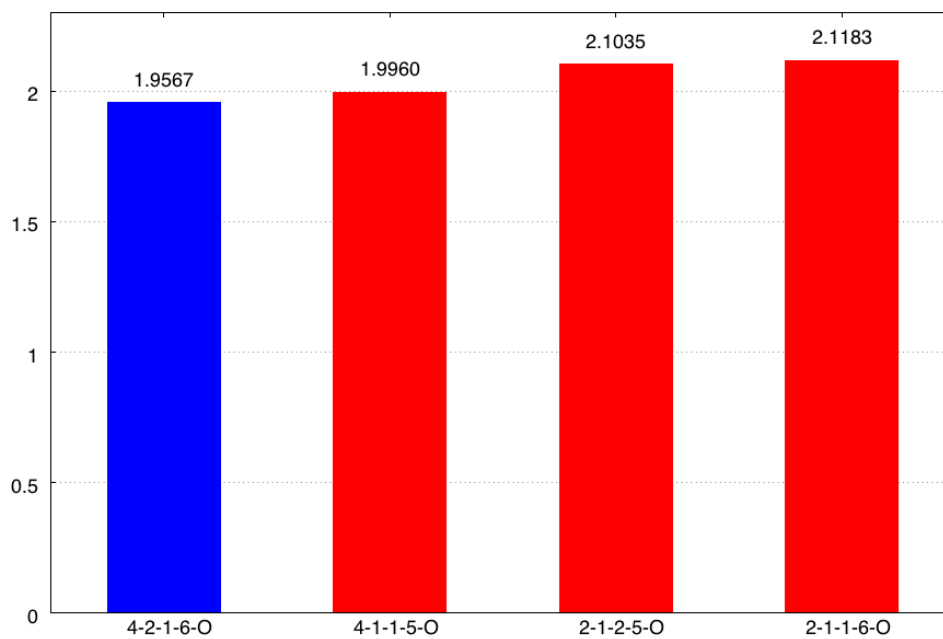
En resumen, el primer experimento demuestra que la fase final de optimización permite mejorar el rendimiento del método original propuesto en el capítulo 3 y el de las versiones *split* propuestas en el capítulo 4.

Tabla 5.1: Primer experimento. Comparativa del rendimiento de las versiones de la optimización *split*. Las versiones se muestran en orden creciente según el valor de **Index WE₂**.

Orden	Versión	μ DP	μ CR	μ E _∞	μ ISE	μ WE ₂	σ WE ₂	μ WE ₂ + σ WE ₂	Index WE ₂
1	4-2-1-6-O	47,8350	29,2153	2,1951	1009,1337	0,8929	1,0638	1,9567	1,0000
2	4-1-1-6-O	47,8307	29,2182	2,1952	1009,5468	0,8927	1,0654	1,9582	1,0008
3	4-2-1-5-O	53,9514	26,4266	1,9023	812,0354	0,8706	1,1215	1,9921	1,0181
4	4-2-2-6-O	46,4064	29,9429	2,2823	1056,2951	0,9102	1,0825	1,9927	1,0184
5	4-1-2-6-O	46,4021	29,9458	2,2824	1056,7083	0,9101	1,0840	1,9941	1,0191
6	4-1-1-5-O	53,9579	26,4255	1,9016	811,7995	0,8710	1,1251	1,9961	1,0201
7	4-2-2-5-O	52,8014	26,9101	1,9568	824,7851	0,8785	1,1314	2,0098	1,0271
8	4-1-2-5-O	52,8021	26,9096	1,9568	824,8208	0,8787	1,1333	2,0121	1,0283
9	4-1-1-2-O	43,5364	31,7053	2,4969	1257,9656	0,9506	1,0955	2,0461	1,0457
10	4-2-1-2-O	43,5279	31,7064	2,4969	1259,0843	0,9513	1,1019	2,0532	1,0493
11	4-1-2-2-O	42,5279	32,3659	2,5728	1298,9540	0,9732	1,1205	2,0937	1,0700
12	2-2-2-5-O	45,7657	30,6860	2,2023	982,8546	0,9263	1,1737	2,1000	1,0732
13	4-2-2-2-O	42,5193	32,3670	2,5728	1300,0727	0,9739	1,1268	2,1007	1,0736
14	2-1-2-5-O	45,7986	30,6597	2,2022	979,7531	0,9263	1,1772	2,1035	1,0750
15	3-2-1-6-O	41,7393	33,0082	2,4629	1203,8732	0,9789	1,1309	2,1097	1,0782
16	3-1-1-6-O	41,7364	33,0075	2,4638	1204,2874	0,9792	1,1323	2,1116	1,0791
17	2-1-1-6-O	39,6843	34,6628	2,6122	1292,7579	1,0014	1,1170	2,1183	1,0826
18	3-2-1-5-O	49,1379	28,9521	2,0528	899,1994	0,9123	1,2063	2,1186	1,0827
19	3-2-2-6-O	40,2921	34,1334	2,5650	1263,8736	0,9930	1,1271	2,1201	1,0835
20	3-1-1-5-O	49,1343	28,9535	2,0532	900,2402	0,9127	1,2087	2,1214	1,0841

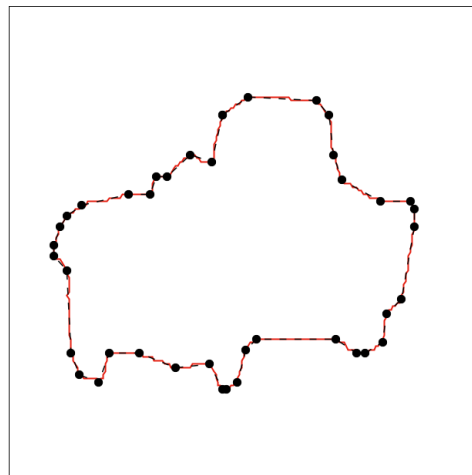


(a) Rendimiento de las versiones *split* optimizadas. Medida $Index WE_2$

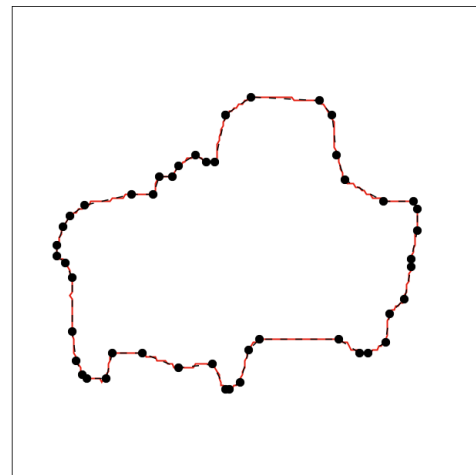


(b) Rendimiento de las versiones *split* optimizadas. Medida $\mu WE_2 + \sigma WE_2$

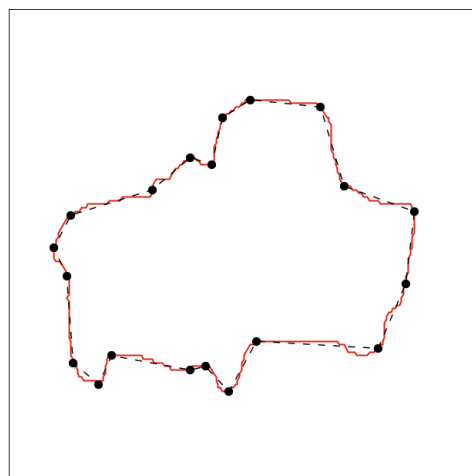
Figura 5.5: Primer experimento. Gráficas comparativas del rendimiento de las versiones *split* optimizadas.



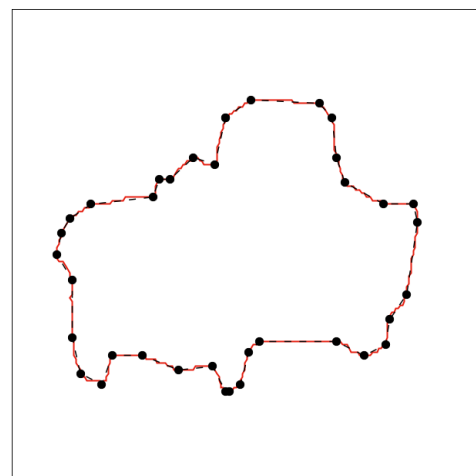
(a) Versión *split* 4-2-1-6 optimizada
DP = 40, $WE_2 = 0,3559$



(b) Versión *split* 4-1-1-5 optimizada
DP = 47, $WE_2 = 0,3805$

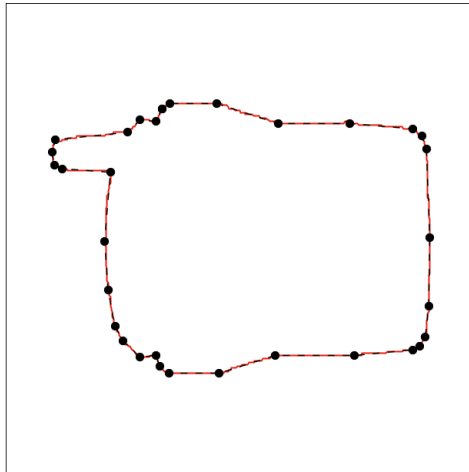


(c) Versión *split* 2-1-1-6 optimizada
DP = 20, $WE_2 = 0,6688$

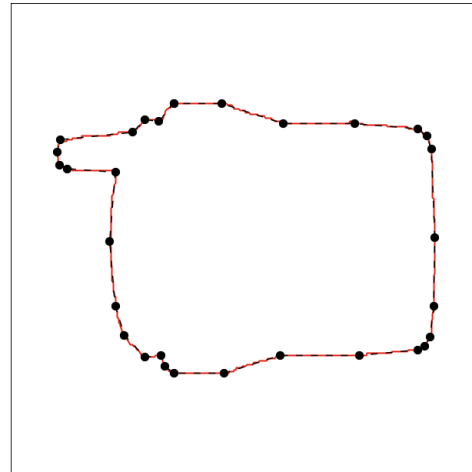


(d) Versión *split* 2-1-2-5 optimizada
DP = 36, $WE_2 = 0,3985$

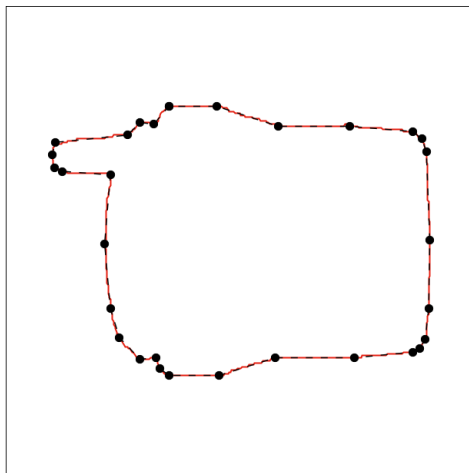
Figura 5.6: Primer experimento: comparación de aproximaciones poligonales del contorno *car-04* ($N = 424$ puntos) obtenidas por versiones optimizadas basadas en la estrategia *split*.



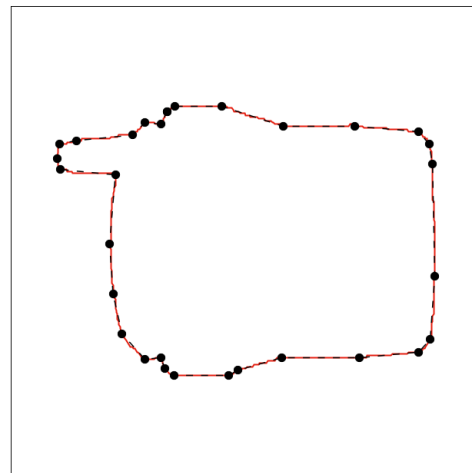
(a) Versión *split* 4-2-1-6 optimizada
DP = 32, $WE_2 = 0,1504$



(b) Versión *split* 4-1-1-5 optimizada
DP = 30, $WE_2 = 0,1635$

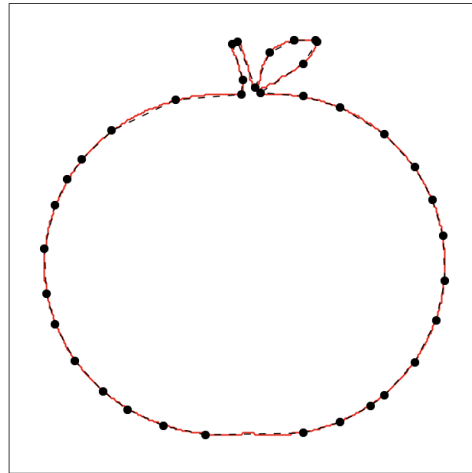


(c) Versión *split* 2-1-1-6 optimizada
DP = 30, $WE_2 = 0,1831$

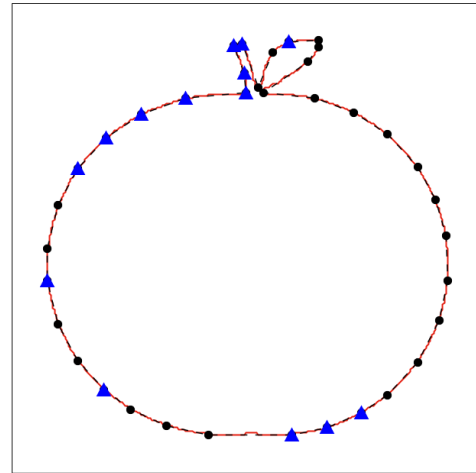


(d) Versión *split* 2-1-2-5 optimizada
DP = 34, $WE_2 = 0,1793$

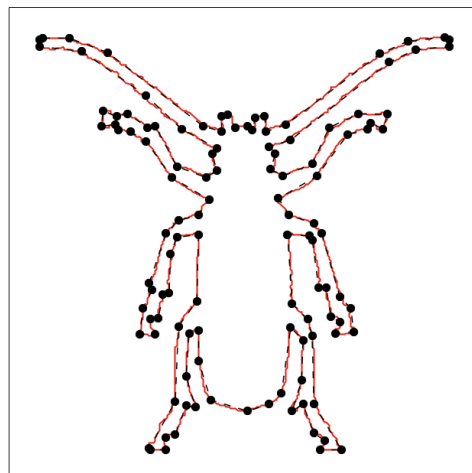
Figura 5.7: Primer experimento: comparación de aproximaciones poligonales del contorno *cellular_phone-17* ($N = 857$ puntos) obtenidas por versiones optimizadas basadas en la estrategia *split*.



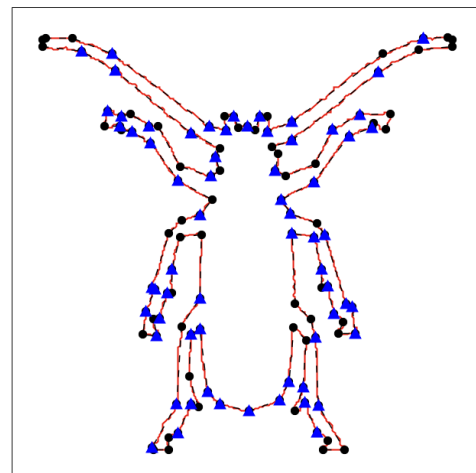
(a) Versión *split* 4-2-1-6
 $DP = 37, WE_2 = 0,5456$



(b) Versión *split* 4-2-1-6 optimizada
 $DP = 37, WE_2 = 0,3143$

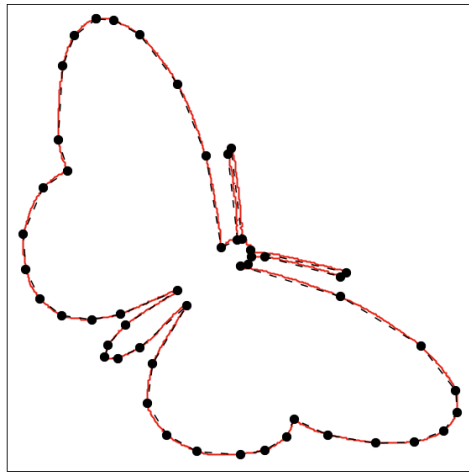


(c) Versión *split* 4-1-1-5
 $DP = 118, WE_2 = 3,4628$

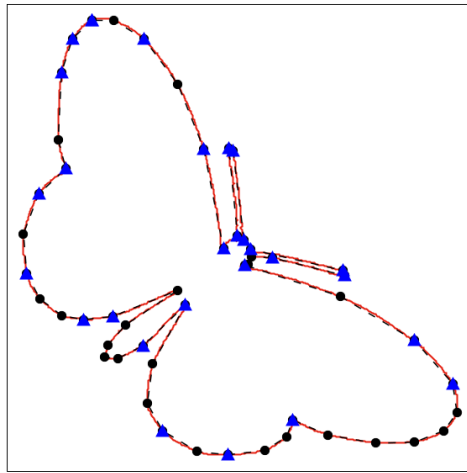


(d) Versión *split* 4-1-1-5 optimizada
 $DP = 118, WE_2 = 1,9722$

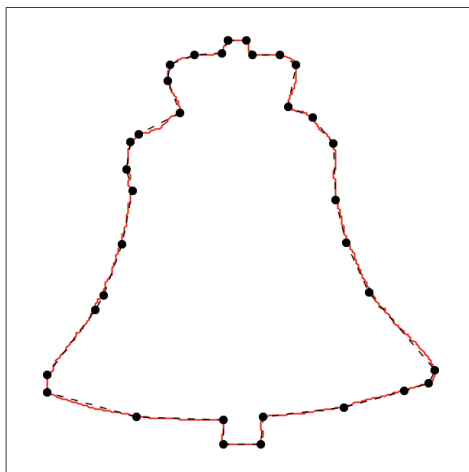
Figura 5.8: Primer experimento: comparativa del ajuste de las versiones *split* 4-2-1-6 y 4-1-1-5 optimizadas respecto a las versiones *split* 4-2-1-6 y 4-1-1-5 originales para los contornos *apple-19_0* ($N = 733$ puntos) y *beetle-20_0* ($N = 2133$ puntos). Nótese que las versiones optimizadas mantienen el mismo número de puntos dominantes, pero estos se han distribuido de forma que se mejora el ajuste poligonal resultante. En el caso del contorno *apple-19_0* se han optimizado 14 puntos y en el caso del contorno *beetle-20_0*, 66 puntos. Los puntos que se han optimizado se han resaltado con un triángulo azul.



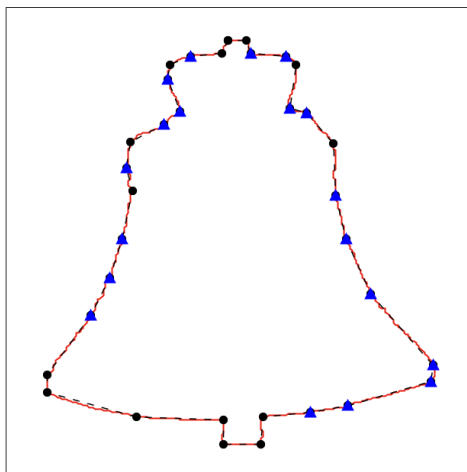
(a) Versión *split 2-1-1-6*
 $DP = 118, WE_2 = 2,1481$



(b) Versión *split 2-1-1-6* optimizada
 $DP = 118, WE_2 = 1,2001$



(c) Versión *split 2-1-2-5*
 $DP = 34, WE_2 = 0,7636$



(d) Versión *split 2-1-2-5* optimizada
 $DP = 34, WE_2 = 0,5158$

Figura 5.9: Primer experimento: comparativa del ajuste de las versiones *split 2-1-1-6* y *2-1-2-5* optimizadas respecto a las versiones *split 2-1-1-6* y *2-1-2-5* originales para los contornos *butterfly-20_0* ($N = 1865$ puntos) y *bell-14_0* ($N = 712$ puntos). Nótese que las versiones optimizadas mantienen el mismo número de puntos dominantes, pero estos se han distribuido de forma que se mejora el ajuste poligonal resultante. En el caso del contorno *butterfly-20_0* se han optimizado 28 puntos y en el caso del contorno *bell-14_0*, 19 puntos. Los puntos que se han optimizado se han resaltado con un triángulo azul.

5.3.2. Segundo experimento: selección de la mejor versión optimizada del método *merge*

Al igual que en el experimento anterior, también se ha añadido el sufijo $-O$ a la notación de las versiones para distinguirlas de las versiones *merge* sin optimizar que se presentaron en el capítulo anterior.

La evaluación cuantitativa de las versiones se recoge en la tabla 5.2 y en la figura 5.10, que muestran el valor medio de las medidas de calidad, la desviación típica de la medida WE_2 , la medida combinada $\mu WE_2 + \sigma WE_2$ y el atributo $Index WE_2$. La tabla 5.2 solo muestra los resultados de las 18 versiones del método optimizado y se encuentra ordenada de forma creciente según el valor de $Index WE_2$ (última columna de la tabla). Un análisis más detallado permite realizar las siguientes observaciones:

■ Posición 1: versión 1-5-O

- Esta es la mejor versión basada en la estrategia *merge* y sus alternativas son:
 - **Medida:** 1.- E_{max} no simétrica
 - **Umbralización:** 5.- método adaptativo
 - **Optimización:** Método de optimización
- Esta versión es la que ha obtenido los mejores resultados al analizar el valor medio $\mu WE_2 = 1,2438$, la desviación típica $\sigma WE_2 = 1,6373$ y la medida combinada $\mu WE_2 + \sigma WE_2 = 2,8812$.
- Esta variante mejora respecto a su versión original, puesto que el valor de la medida combinada $\mu WE_2 + \sigma WE_2$ de la versión *merge* 1-5 es de 6,2794 y la versión *merge* 1-5-O optimizada posee un valor de 2,8812 (véanse las tablas 4.4 y 5.2).

■ Posición 2: versión 2-5-O

- Alternativas de esta versión:
 - **Medida:** 2.- ISE no simétrica
 - **Umbralización:** 5.- método adaptativo
 - **Optimización:** Método de optimización
- Esta versión ha obtenido un valor medio $\mu WE_2 = 1,4887$, la desviación típica $\sigma WE_2 = 1,6532$. El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 3,1420 y su $Index WE_2 = 1,0905$.
- Esta variante mejora respecto a su versión original, puesto que el valor de la medida combinada $\mu WE_2 + \sigma WE_2$ de la versión *merge* 2-5 es de 5,5806 y la versión *merge* 2-5-O optimizada posee un valor de 3,1420 (véanse las tablas 4.4 y 5.2).

■ **Posición 3:** versión *1-6-O*

- Alternativas de esta versión:
 - **Medida:** 1.- E_{max} no simétrica
 - **Umbralización:** 6.- método semi-adaptativo
 - **Optimización:** Método de optimización
- Esta versión ha obtenido un valor medio $\mu WE_2 = 1,4401$, la desviación típica $\sigma WE_2 = 1,7729$. El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 3,2130 y su $Index WE_2 = 1,1151$.
- Esta variante mejora respecto a su versión original, puesto que el valor de la medida combinada $\mu WE_2 + \sigma WE_2$ de la versión merge 1-6 es de 7,6355 y la versión merge 1-6-O optimizada posee un valor de 3,2130 (véanse las tablas 4.4 y 5.2).

■ **Posición 4:** versión *2-6-O*

- Alternativas de esta versión:
 - **Medida:** 2.- ISE no simétrica
 - **Umbralización:** 6.- semi-adaptativo
 - **Optimización:** Método de optimización
- Esta versión ha obtenido un valor medio $\mu WE_2 = 1,6746$, la desviación típica $\sigma WE_2 = 1,8470$. El valor obtenido para la medida combinada $\mu WE_2 + \sigma WE_2$ es 3,5216 y su $Index WE_2 = 1,2223$.
- Esta variante mejora respecto a su versión original, puesto que el valor de la medida combinada $\mu WE_2 + \sigma WE_2$ de la versión merge 1-6 es de 6,2265 y la versión merge 1-6-O optimizada posee un valor de 3,5216 (véanse las tablas 4.4 y 5.2).

Las figuras 5.11 y 5.12 permiten realizar una evaluación cualitativa de las aproximaciones poligonales generadas por las cuatro mejores alternativas basadas en la estrategia *merge* (*1-5-O*, *2-5-O*, *1-6-O* y *2-6-O*) para dos contornos de la base de datos de pruebas *MPEG-7*. Además, en las figuras 5.13 y 5.14 se muestra una comparativa de las versiones *1-5-O*, *2-5-O*, *1-6-O* y *2-6-O* respecto a sus versiones originales (sin optimizar).

Por último, La tabla 5.3 permite comparar los resultados de la mejor versión basada en la estrategia *merge* (versión *1-5-O*) con los resultados de las versiones basadas en la estrategia *split* (*4-2-1-6-O*, *4-1-1-5-O*, *2-1-1-6-O* y *2-1-2-5-O*) que fueron analizadas en el primer experimento (véase la sección 5.3.2). Las figuras 5.15 y 5.16 muestran algunas de las aproximaciones poligonales generadas por estas versiones para facilitar una evaluación cualitativa.

El análisis de los resultados de la tabla 5.3 permite realizar la siguiente observación:

- La versión *1-5-O* es la que ha obtenido los mejores resultados entre todas las basadas en la estrategia *merge*. Sin embargo, no consigue superar a las versiones basadas en la estrategia *split* ni a la versión correspondiente al método original propuesto en el capítulo 3. En concreto, la versión *1-5* del método *merge* ocupa la posición número 73 del ranking si se la compara con todas las demás versiones de la estrategia *split*, con un valor *Index WE₂* de 1,4725.

Sin embargo, con este segundo experimento se demuestra que la estrategia *merge* optimizada no logra obtener un mejor rendimiento respecto a las versiones *split* optimizadas propuestas en el experimento anterior (versiones *4-2-1-6-O* y *4-1-1-6-O*).

5.4. Conclusiones

En este capítulo se ha implementado una etapa final de optimización sobre las versiones de los métodos *split* y *merge* presentadas en el capítulo anterior. Los métodos *split* y *merge* optimizados constan de ocho y siete etapas, respectivamente. Ambos métodos poseen la misma cantidad de versiones propuestas en la parametrización: 288 versiones en el caso de la estrategia *split* y 18 en el caso de la estrategia *merge*. La complejidad computacional de las versiones optimizadas es $\mathcal{O}(n \log(n))$.

De los experimentos realizados para evaluar el rendimiento se concluye que:

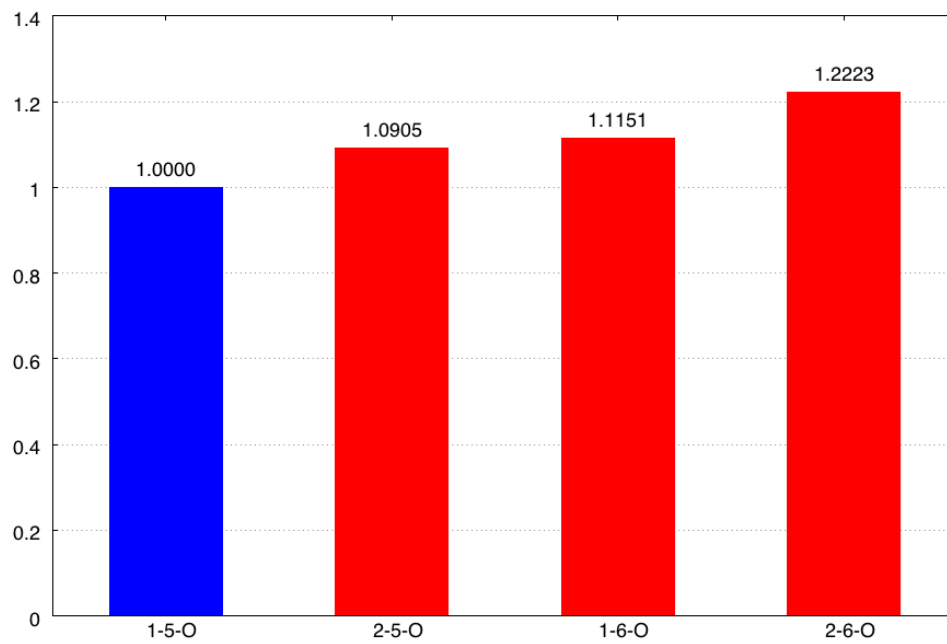
- Hay 13 versiones basadas en la estrategia *split* que han obtenido mejores resultados que la versión *2-1-2-5-O*, que se corresponde con el método original optimizado.
- La versión *4-2-1-6-O*, que hace uso del *casco convexo*, es la que ha obtenido los mejores resultados entre este tipo de versiones. Sin embargo, esta versión requiere la aplicación de una fase final de supresión de puntos *superfluos* que están *casi alineados*. La versión *4-1-1-6-O* obtiene resultados similares a la versión *4-2-1-6-O*.
- De todas las versiones *merge*, la versión *1-5-O* ha obtenido los mejores resultados, pero no supera al por el método original (*2-1-2-5-O*) ni a las mejores versiones *split* (*4-2-1-6-O* y *4-1-1-6-O*).

Se considera que el proceso de optimización ha tenido éxito porque ha permitido definir dos versiones más eficientes del método (*4-2-1-6-O* y *4-1-1-6-O*):

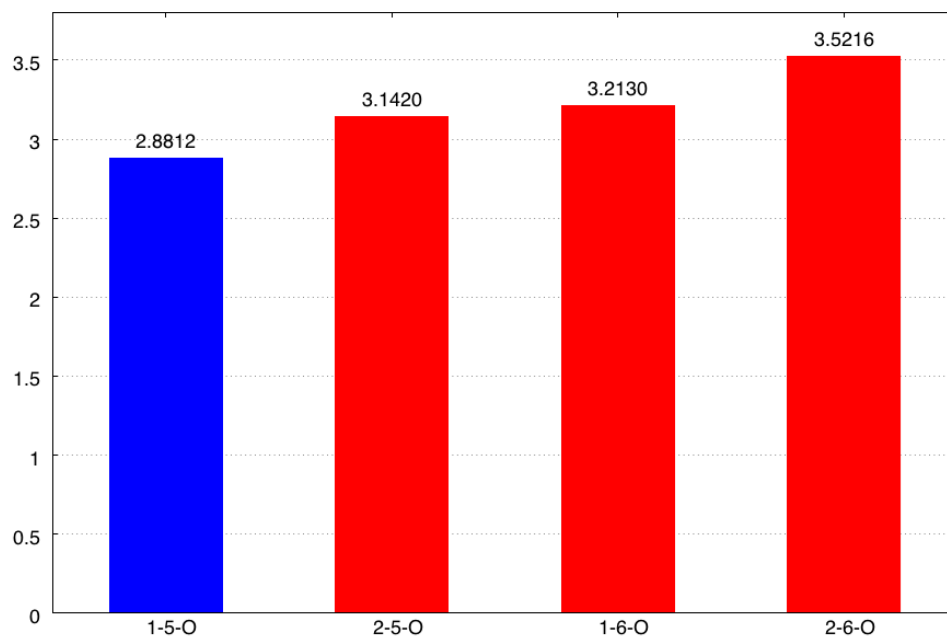
- **Selección de iniciales:** 4.- Casco convexo
- **Medida y simetría:** 1.- Emax *simétrica* o 2.- Emax *no simétrica*
- **Significación de iniciales:** 1.- Distancia máxima al centroide
- **Umbralización:** 6.- Método semi-adaptativo
- **Optimización:** Método de optimización

Tabla 5.2: Segundo experimento. Comparativa del rendimiento de las versiones de la optimización *merge*. Las versiones se muestran en orden creciente según el valor de **Index WE₂**.

Orden	Versión	μ DP	μ CR	μ E ∞	μ ISE	μ WE ₂	σ WE ₂	μ WE ₂ + σ WE ₂	Index WE ₂
1	1-5-O	43,0950	27,5188	2,6673	1387,2628	1,2438	1,6374	2,8812	1,0000
2	2-5-O	26,1271	50,5515	4,8544	5166,9100	1,4887	1,6532	3,1420	1,0905
3	1-6-O	33,8986	34,8433	3,4430	2211,4747	1,4401	1,7729	3,2130	1,1151
4	2-6-O	23,2814	56,0000	5,6034	6958,6238	1,6746	1,8470	3,5216	1,2223
5	1-2-O	29,5907	40,1021	4,1062	3134,6101	1,6154	1,9577	3,5731	1,2401
6	1-1-O	28,7300	41,1158	4,3266	3560,3325	1,6953	2,0821	3,7775	1,3111
7	2-2-O	21,6236	59,9894	6,2094	8781,6952	1,8219	1,9784	3,8002	1,3190
8	1-4-O	27,1371	43,5057	4,7811	4340,5392	1,8217	2,2034	4,0251	1,3970
9	2-1-O	19,5607	65,4405	7,3043	12200,6421	2,1084	2,2936	4,4020	1,5278
10	3-5-O	37,2464	38,8720	4,1468	3320,3077	1,9979	2,4771	4,4749	1,5531
11	2-3-O	18,2543	70,7298	7,9819	14435,8170	2,1896	2,3303	4,5199	1,5688
12	3-6-O	31,6514	44,9230	4,9433	4851,2360	2,1793	2,5088	4,6882	1,6272
13	1-3-O	21,6243	57,0679	6,5945	9056,3766	2,2456	2,5233	4,7689	1,6552
14	3-2-O	28,6586	49,4161	5,5587	6258,2408	2,3560	2,7428	5,0988	1,7697
15	2-4-O	17,5300	72,3227	8,8152	17741,5413	2,4470	2,6760	5,1230	1,7781
16	3-1-O	27,2029	51,5670	5,9842	7397,9102	2,5830	2,9705	5,5534	1,9275
17	3-4-O	25,6964	55,2423	6,6357	8924,8745	2,7985	3,2132	6,0117	2,0865
18	3-3-O	22,7121	63,0797	7,5690	12305,5353	2,8402	3,2156	6,0558	2,1018

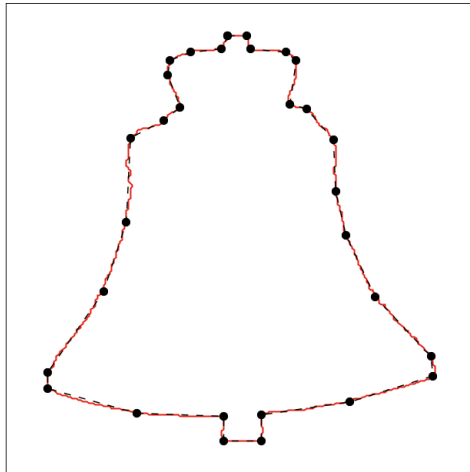


(a) Rendimiento de las versiones *merge* optimizadas. Medida $Index WE_2$

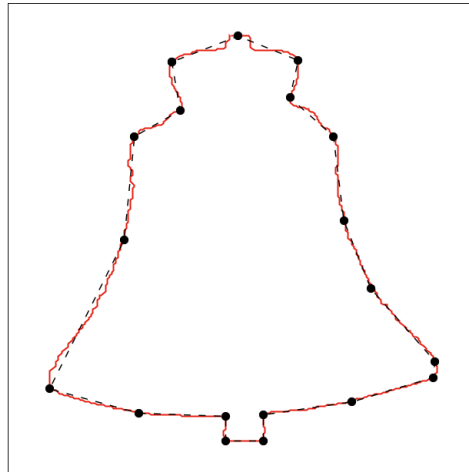


(b) Rendimiento de las versiones *merge* optimizadas. Medida $\mu WE_2 + \sigma WE_2$

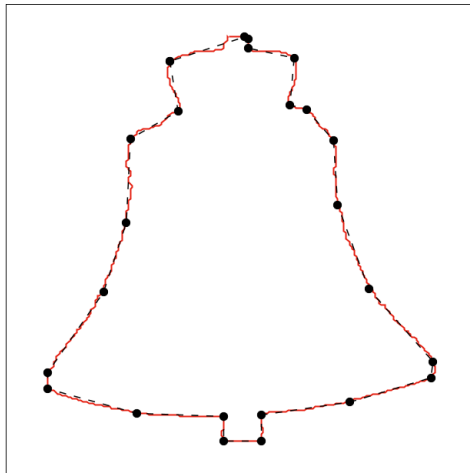
Figura 5.10: Segundo experimento. Gráficas comparativas del rendimiento de las versiones *merge* optimizadas.



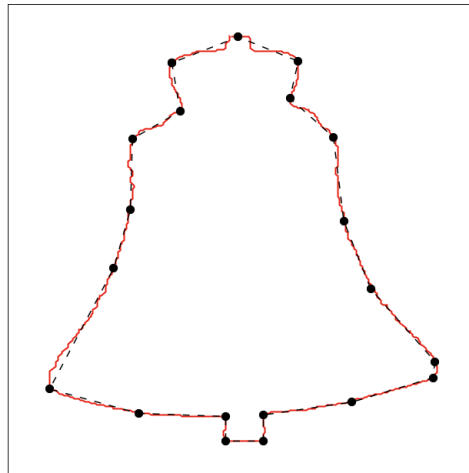
(a) Versión *merge 1-5* optimizada
DP = 30, $WE_2 = 0,4781$



(b) Versión *merge 2-5* optimizada
DP = 19, $WE_2 = 0,7377$



(c) Versión *merge 1-6* optimizada
DP = 24, $WE_2 = 0,5553$



(d) Versión *merge 2-6* optimizada
DP = 20, $WE_2 = 0,7164$

Figura 5.11: Segundo experimento: comparación de aproximaciones poligonales del contorno *bell-14_0* ($N = 712$ puntos) obtenidas por versiones optimizadas basadas en la estrategia *merge*.

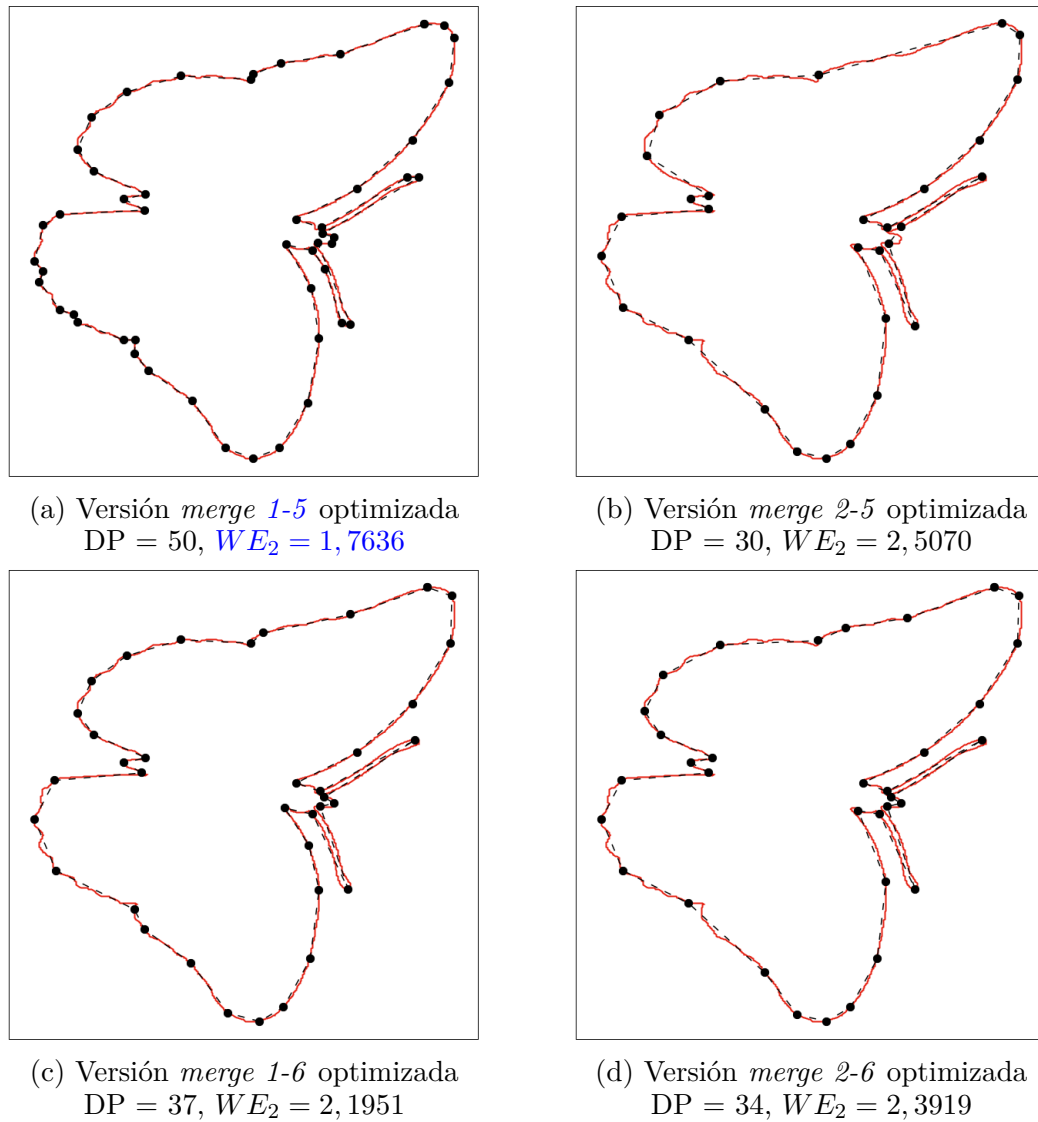
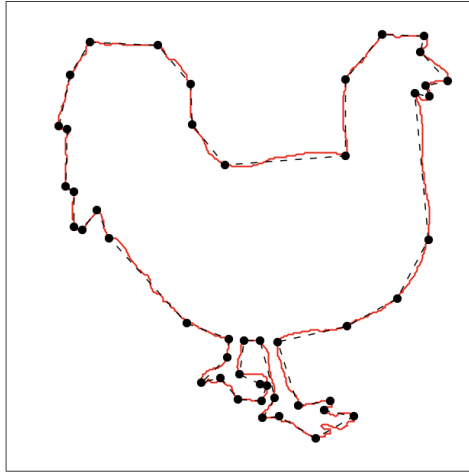
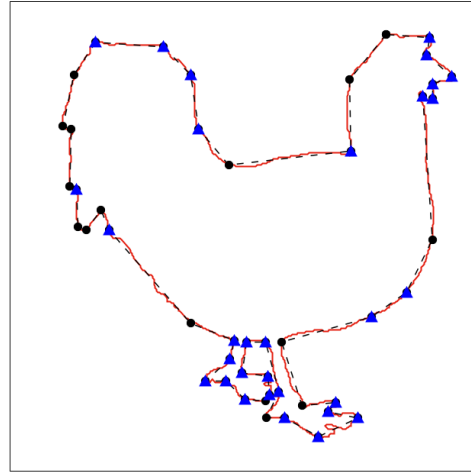


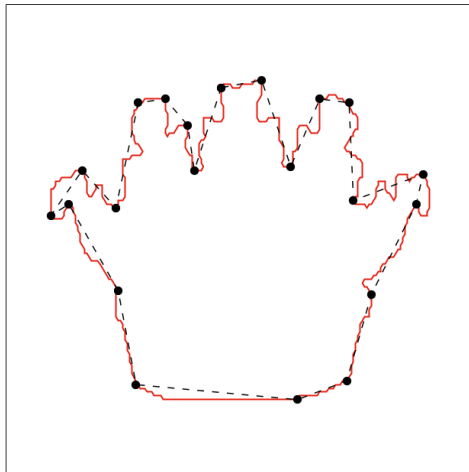
Figura 5.12: Segundo experimento: comparación de aproximaciones poligonales del contorno *butterfly-16_0* ($N = 1819$ puntos) obtenidas por versiones optimizadas de la estrategia *merge*.



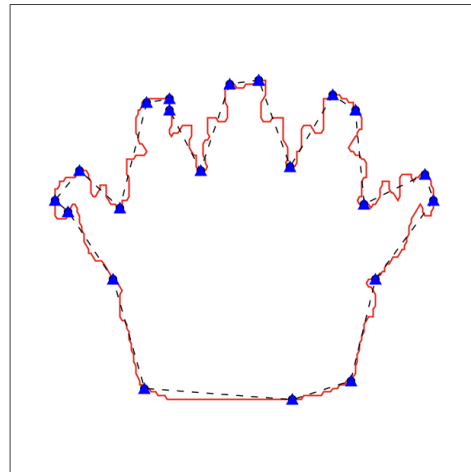
(a) Versión *merge 1-5*
 DP = 47, $WE_2 = 4,6280$



(b) Versión *merge 1-5* optimizada
 DP = 47, $WE_2 = 2,6782$



(c) Versión *merge 2-5*
 DP = 21, $WE_2 = 3,9416$



(d) Versión *merge 2-5* optimizada
 DP = 21, $WE_2 = 2,3195$

Figura 5.13: Segundo experimento: comparativa del ajuste de las versiones *merge 1-5* y *2-5* optimizadas respecto a las versiones *merge 1-5* y *2-5* originales para los contornos *chicken-15_0* ($N = 1304$ puntos) y *crown-10_0* ($N = 570$ puntos). Nótese que las versiones optimizadas mantienen el mismo número de puntos dominantes, pero estos se han distribuido de forma que se mejora el ajuste poligonal resultante. En el caso del contorno *chicken-15_0* se han optimizado 31 puntos y en el caso del contorno *crown-10_0*, 22 puntos. Los puntos que se han optimizado se han resaltado con un triángulo azul.

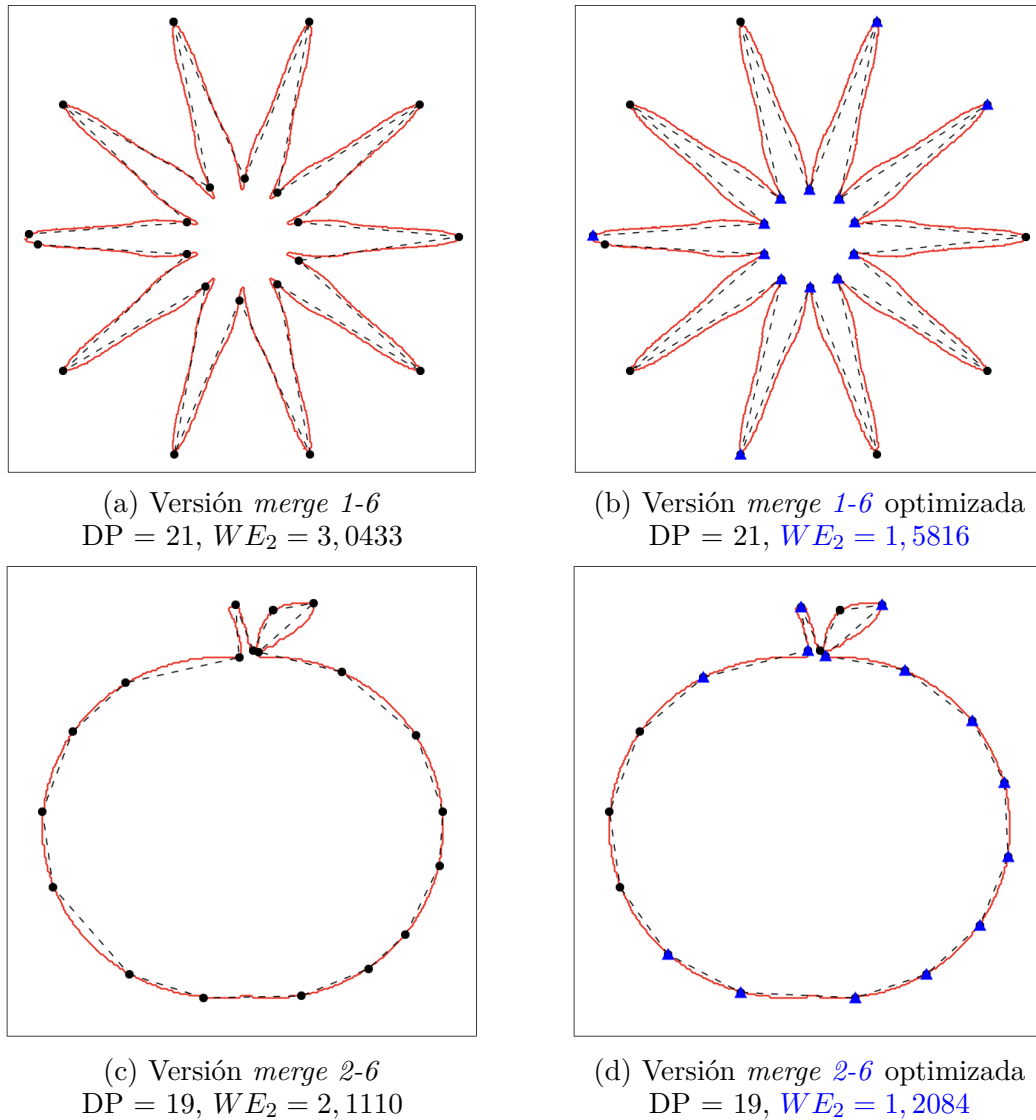


Figura 5.14: Segundo experimento: comparativa del ajuste de las versiones *merge 1-6* y *2-6* optimizadas respecto a las versiones *merge 1-6* y *2-6* originales para los contornos *device7-8_0* ($N = 3803$ puntos) y *apple-19_0* ($N = 733$ puntos). Nótese que las versiones optimizadas mantienen el mismo número de puntos dominantes, pero estos se han distribuido de forma que se mejora el ajuste poligonal resultante. En el caso del contorno *device7-8_0* se han optimizado 14 puntos y en el caso del contorno *apple-19_0*, 14 puntos. Los puntos que se han optimizado se han resaltado en azul y se han marcado con una forma triangular.

Tabla 5.3: Comparativa del rendimiento de las mejores versiones *split* y *merge*. Las versiones se muestran en orden creciente según el valor de **Index WE₂**.

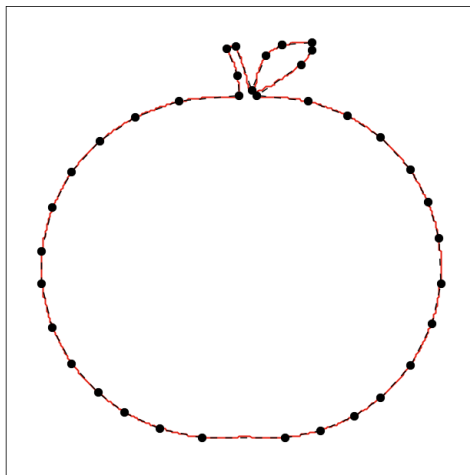
Estrategia	Versión	μ DP	μ CR	μ E _{max}	μ ISE	μ WE ₂	σ WE ₂	μ WE ₂ + σ WE ₂	Index WE ₂
Split (1)	4-2-1-6-O	47,8350	29,2152	2,1951	1009,1336	0,8929	1,0638	1,9567	1,0000
Split (6)	4-1-1-5-O	53,9578	26,4254	1,9016	811,7995	0,8710	1,1251	1,9961	1,0201
Split (14) (*)	2-1-2-5-O	45,7986	32,3670	2,5728	1300,0727	0,9739	1,1268	2,1007	1,0736
Split (17)	2-1-1-6-O	39,6843	34,6628	2,6122	1292,7579	1,0014	1,1170	2,1183	1,0826
Merge (1)	1-5-O	43,0950	27,5188	2,6673	1387,2628	1,2438	1,6373	2,8812	1,4725

Notas:

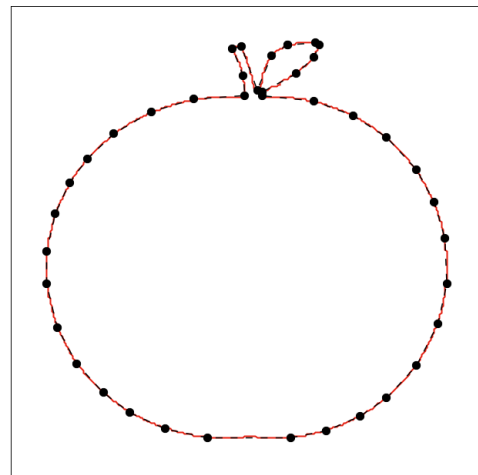
Entre paréntesis se muestra la posición de cada versión en las tablas originales (5.1 y 5.2)

(*) La versión *2-1-2-5* representa al método *split* presentado en el capítulo 3.

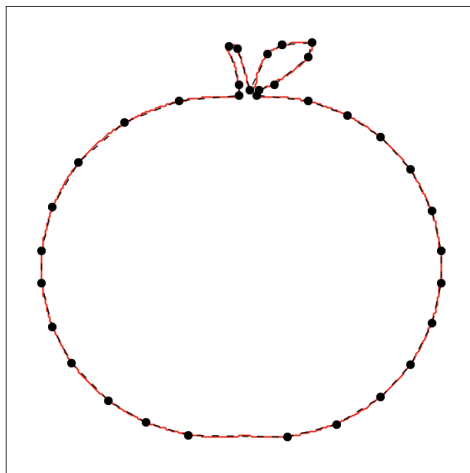
El valor de **Index WE₂** ha sido recalculado para las versiones que se muestran en la tabla.



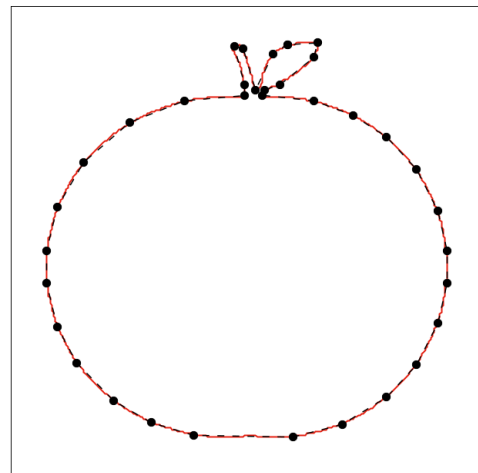
(a) Versión *split* 4-2-1-6 optimizada
 DP = 37, $WE_2 = 0,3143$



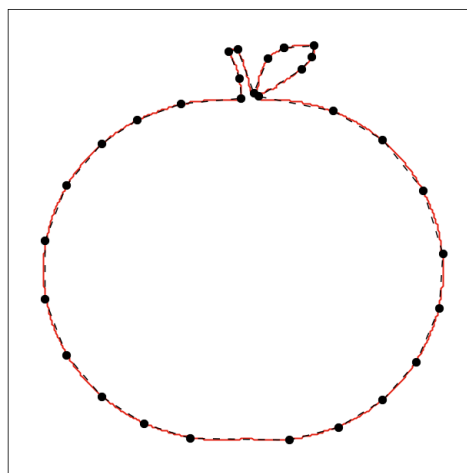
(b) Versión *split* 4-1-1-5 optimizada
 DP = 40, $WE_2 = 0,3244$



(c) Versión *split* 2-1-1-6 optimizada
 DP = 35, $WE_2 = 0,3811$

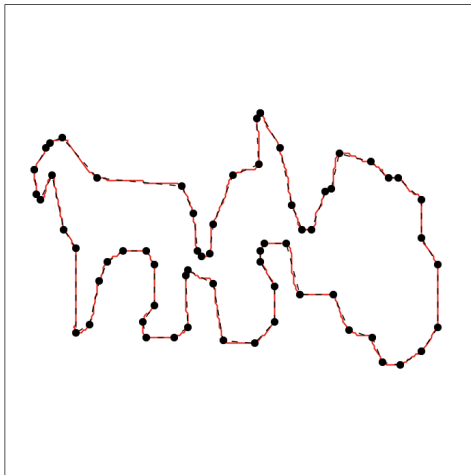


(d) Versión *split* 2-1-2-5 optimizada
 DP = 35, $WE_2 = 0,3811$

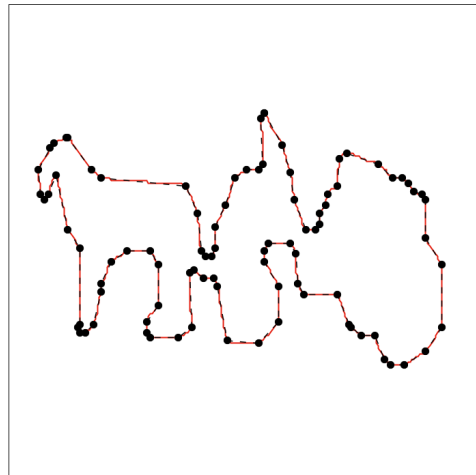


(e) Versión *merge* 1-5 optimizada
 DP = 30, $WE_2 = 0,5476$

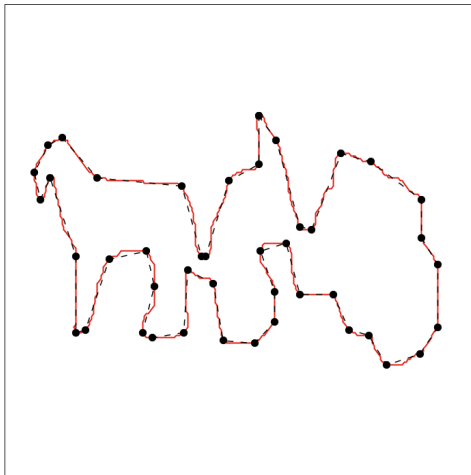
Figura 5.15: Comparación de aproximaciones poligonales del contorno del contorno *apple-19* ($N = 733$ puntos) obtenidas por versiones optimizadas de las estrategias *split* y *merge*.



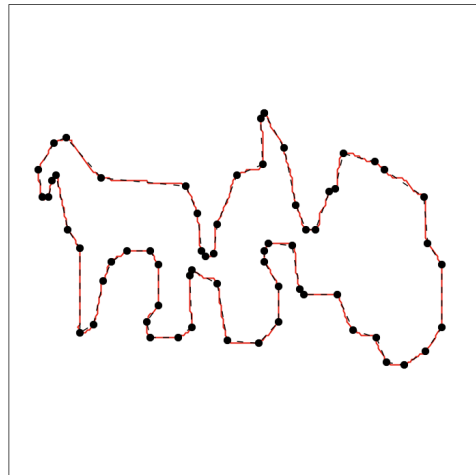
(a) Versión *split* 4-2-1-6 optimizada
 DP = 64, $WE_2 = 0,8594$



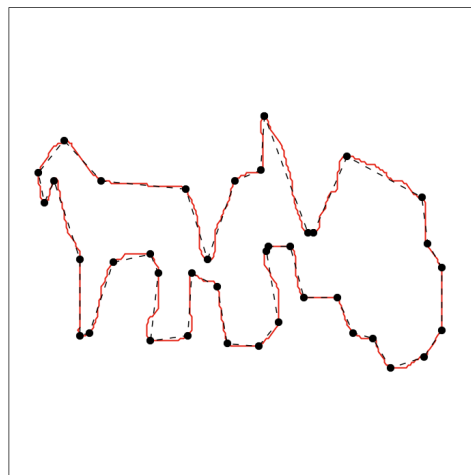
(b) Versión *split* 4-1-1-5 optimizada
 DP = 91, $WE_2 = 0,8670$



(c) Versión *split* 3-2-1-5 optimizada
 DP = 44, $WE_2 = 1,3749$



(d) Versión *split* 2-1-2-5 optimizada
 DP = 64, $WE_2 = 0,8926$



(e) Versión *merge* 1-5 optimizada
 DP = 39, $WE_2 = 1,7654$

Figura 5.16: Comparación de aproximaciones poligonales del contorno del contorno *device9-1* ($N = 742$ puntos) obtenidas por versiones optimizadas de las estrategias *split* y *merge*.

6

Conclusiones y futuros estudios

6.1. Conclusiones

El desarrollo de esta Tesis Doctoral ha permitido proponer un nuevo método automático que genera aproximaciones poligonales de contornos cerrados de objetos bidimensionales. Como consecuencia del trabajo desarrollado, se han obtenido las siguientes conclusiones:

1. En primer lugar, se ha desarrollado un nuevo método *split* que genera aproximaciones poligonales:
 - Se basa en el concepto de umbralización unimodal de los niveles de significación de los puntos del contorno.
 - Se compone de seis etapas.
 - Se ha comprobado que obtiene mejores resultados que la versión *split* automatizada del método de *Ramer, Douglas–Peucker* (RDP) [52, 84] y que las versiones *merge* automatizadas de los métodos de *Carmona* [53] y *Masood* [17, 57], presentados por *Prasad et al.* [76].
2. Se ha desarrollado un análisis exhaustivo del método propuesto:
 - Este estudio se ha basado en el análisis de las estrategias *split* y *merge*.
 - En el estudio se han definido 288 versiones *split* y 18 versiones *merge*.
 - Los experimentos han demostrado que una versión *split*, basada en el uso del casco convexo y que emplea la medida E_{max} y el método de umbralización semi-

adaptativo, ha obtenido los mejores resultados, superando incluso al nuevo método propuesto.

3. Por último, se ha completado el análisis incorporando una etapa final de optimización:
 - El método de optimización se basa en el método propuesto por Masood [57].
 - Las pruebas desarrolladas demuestran que el método optimizado mejora a todas las versiones *split* y *merge* propuestas, mejorando también al método original de forma significativa.

En resumen, el desarrollo de esta Tesis Doctoral ha permitido obtener un método heurístico que genera aproximaciones poligonales eficientes. Estas aplicaciones pueden ser utilizadas en aplicaciones de tiempo real, superando las dificultades que presentan los algoritmos óptimos, que requieren de una carga computacional mayor.

6.2. Futuros estudios

Como consecuencia de los resultados obtenidos en esta Tesis Doctoral, se han determinado dos aspectos que son susceptibles de ser abordados con nuevas propuestas:

1. **Combinación de las estrategias *split* y *merge*:** la combinación de ambas estrategias podría desarrollar un método que permita generar aproximaciones poligonales haciendo uso de las ventajas de cada una de ellas.
2. **Estudio de las medidas de calidad:** en la presente Tesis Doctoral se ha utilizado fundamentalmente la medida WE_2 , pues se considera que es la más adecuada entre todas las propuestas en la literatura para evaluar la calidad de las aproximaciones poligonales. Sin embargo, de un examen detenido de la literatura específica se desprende que esta medida tampoco refleja exactamente la percepción humana sobre la calidad de una aproximación poligonal. En función de ello, este tópico aún permanece abierto, susceptible de proponer, en su caso, una nueva medida de calidad que permita evaluar de forma objetiva las aproximaciones poligonales.



A new thresholding approach for automatic generation of polygonal approximations

Este artículo presenta un método para la obtención de aproximaciones poligonales basado en el algoritmo de Ramer, Douglas-Peucker (método RDP). Este artículo ha sido publicado en la revista *Journal of Visual Communication and Image Representation* en el año 2016 [96].

- **Artículo:** N.L. Fernández-García, L. Del-Moral Martínez, A. Carmona-Poyato, F.J. Madrid-Cuevas y R. Medina Carnicer. *A new thresholding approach for automatic generation of polygonal approximations*. *Journal of Visual Communication and Image Representation*. Volumen 35. 155-168. ISSN: 1047-3203. Doi: 10.1016/j.jvcir.2015.12.013.
- **Revista:** *Journal of Visual Communication and Image Representation*
 - **Índice de impacto:** JCR 2015: 1.530
 - **Categoría:** Computer science, software engineering.
 - **Posición:** puesto 19 de 105
 - **Cuartil:** Q1



A new thresholding approach for automatic generation of polygonal approximations[☆]



N.L. Fernández-García^{*}, L. Del-Moral Martínez, A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Medina-Carnicer

Department of Computing and Numerical Analysis, Maimonides Institute for Biomedical Research (IMIBIC), University of Córdoba, Córdoba, Spain

ARTICLE INFO

Article history:

Received 18 July 2015

Accepted 18 December 2015

Available online 29 December 2015

Keywords:

Digital planar curves

Polygonal approximation

Dominant points

Significance of contour points

Thresholding techniques

Object recognition

Assessment of polygonal approximations

Corner points

Automatic algorithm

ABSTRACT

The present paper proposes a new algorithm for automatic generation of polygonal approximations of 2D closed contours based on a new thresholding method. The new proposal computes the significance level of the contour points using a new *symmetric* version of the well-known Ramer, Douglas–Peucker method, and then a new *Adaptive* method is applied to *threshold* the normalized significance level of the contour points to generate the polygonal approximation. The experiments have shown that the new algorithm has good performance for generating polygonal approximations of 2D closed contours. Furthermore, the new algorithm does not require any parameter to be tuned.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Shape representation by polygonal approximation has been extensively used for constructing a characteristic description of a boundary in the form of a series of straight lines. This representation is very popular due its simplicity, locality, generality, and compactness [17,24,40]. In a closed digital planar curve, most of the information is located at points of high curvature [3], which are used to obtain polygonal approximations. These points are known as *dominant points* and are an important target in many machine vision applications [39].

The polygonal approximation of closed digital planar curves, also called *contours*, can be formulated in two ways [13]:

- *Minimum-distortion problem* or *Min – ϵ problem*: these methods are based on a predefined number N of vertices and try to generate the *optimal* polygonal approximation with N vertices or dominant points (*DP*) so that its distance from the contour is minimal among all the approximations with N vertices.

- *Minimum-rate problem* or *Min – # problem*: these methods are based on a predefined error measure ϵ and try to generate the polygonal approximations, with the minimal number of vertices, that distant from the contour by no more than ϵ .

Many methods have been proposed to construct polygonal approximations of closed digital planar curves. These methods can be classified into two major categories:

- *Optimization approach*: these methods are based on an optimization criterion [1,13,28,29,34,35].
- *Heuristic approach*: these methods construct the approximating polygon using some reasonable geometric or perceptual features of the contours [10]
 - *Parametric approach*: these methods take into account a parameter to generate the polygonal approximations [2,4,5,7,18,21–23].
 - *Non-parametric approach*: these algorithms generate the polygonal approximations without using parameters [19,20,30,37,39].

The optimization methods have two drawbacks [10]: although they find good solutions, the optimum depends on the applied criterion and requires a very high computational complexity, which is not suitable for real time applications. On the contrary,

[☆] This work has been developed with the support of the Research Projects called TIN2012-32952 and BROCA both financed by Economy and Competitiveness Ministry of Spain and FEDER.

^{*} Corresponding author. Fax: +34 957218630.

E-mail addresses: ma1fegan@uco.es (N.L. Fernández-García), i52momal@uco.es (L. Del-Moral Martínez), ma1capoa@uco.es (A. Carmona-Poyato), ma1macuf@uco.es (F.J. Madrid-Cuevas), rmedina@uco.es (R. Medina-Carnicer).

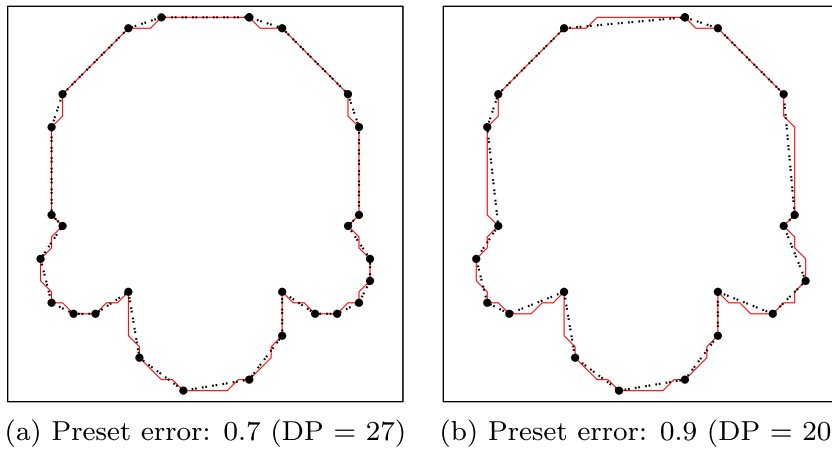


Fig. 1. Drawbacks of the original RDP algorithm: dependence on a parametric error and lack of symmetry. The figure shows the polygonal approximations generated with two preset errors and the number of dominant points (DP).

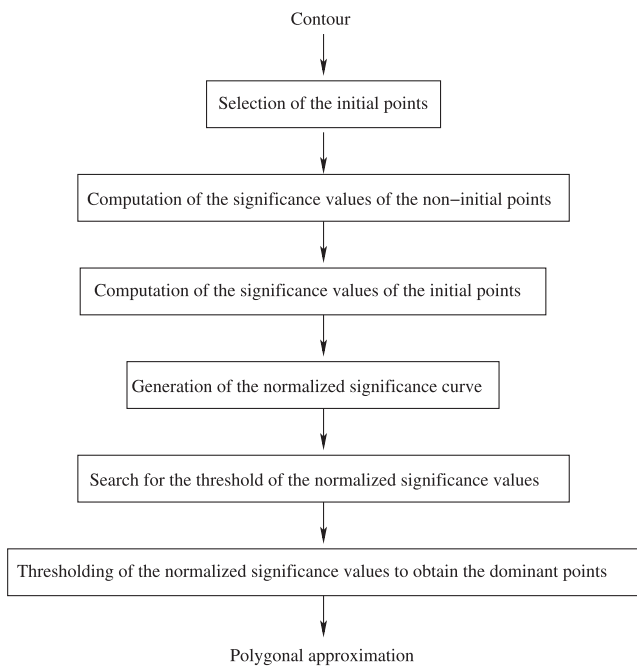


Fig. 2. Framework of the proposed algorithm for polygonal approximation.

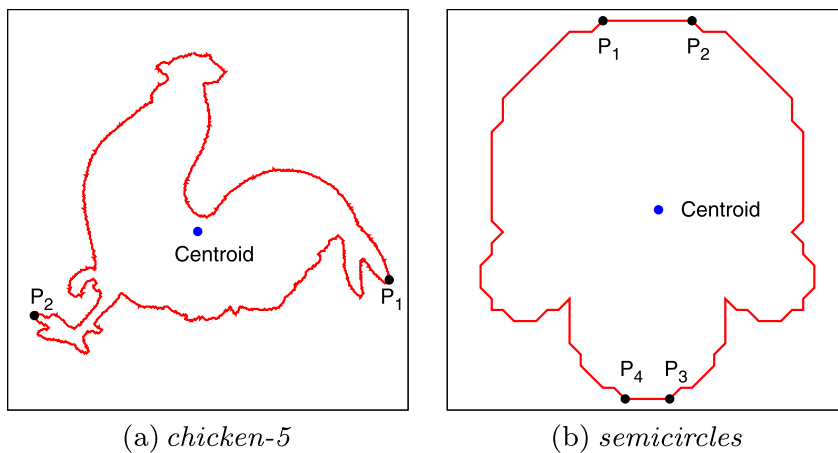


Fig. 3. First step - Selection of initial points for the contour: (a) P_1 is the farthest point to the centroid and P_2 is the farthest point to P_1 and (b) P_1 and P_2 are the farthest points to the centroid and P_3 and P_4 are the farthest points to the previous ones, respectively.

the heuristic methods do not guarantee any kind of optimum; however, these algorithms can find reasonable polygonal approximations for real time applications.

Real images show irregular shapes and a lot of variability and different levels of detail. Only an algorithm working independently of input parameters, adjusting itself to each curve, could give satisfactory results if an efficient and reliable representation of the contours is desired [11]. The need for reliable and robust non-parametric algorithms that can be applied to different curves satisfactorily is still an issue of high importance [19].

The present work proposes a new *automatic* and *non-parametric* method for polygonal approximations which is based on a new *symmetric* version of the well-known Ramer, Douglas–Peucker method [7,31] and applies a new *Adaptive* thresholding method to obtain the dominant points. The present paper is arranged as follow. Section 2 describes a related work. Section 3 explains the new proposal. The experiments and results are described in Section 4. Finally, the main conclusions are summarized in Section 5.

2. Related work

The proposed algorithm is based on a modification of the Ramer, Douglas–Peucker method (RDP method) [7,31], which is a scale-independent split algorithm for subdividing a curve into its

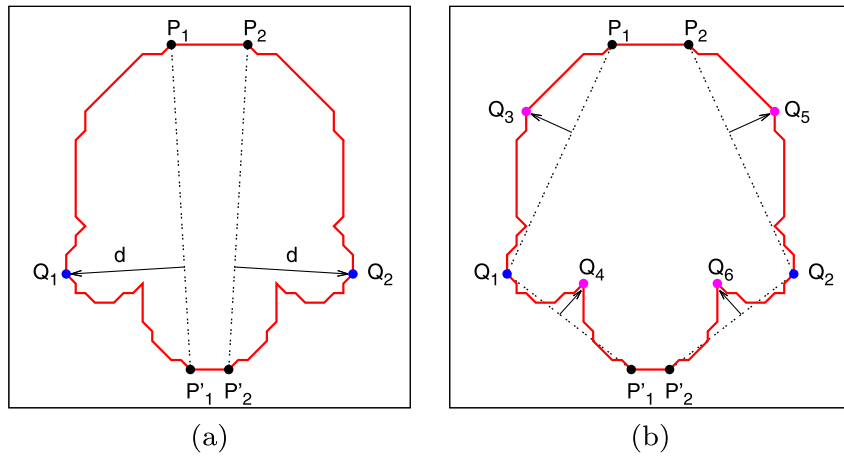


Fig. 4. Second step – Computation of the significance of the non-initial points: performance of the recursive process *compute_significance_values* on the contour *semicircles*: (a) 2 first candidate points Q_1 and Q_2 and their significance value d and (b) 4 following candidate points: Q_3 – Q_6 .

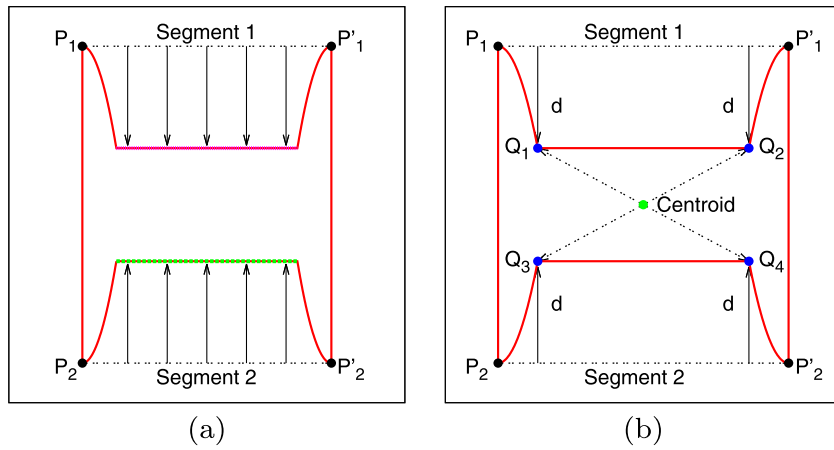


Fig. 5. Second step – Computation of the significance of the non-initial points: contour with straight sides (a) farthest points to the segments defined by the *initial points* P_1, P'_1, P_2, P'_2 and (b) farthest points to the centroid among the previous farthest points to the segments: d is the significance value of the candidate points Q_1 – Q_4 .

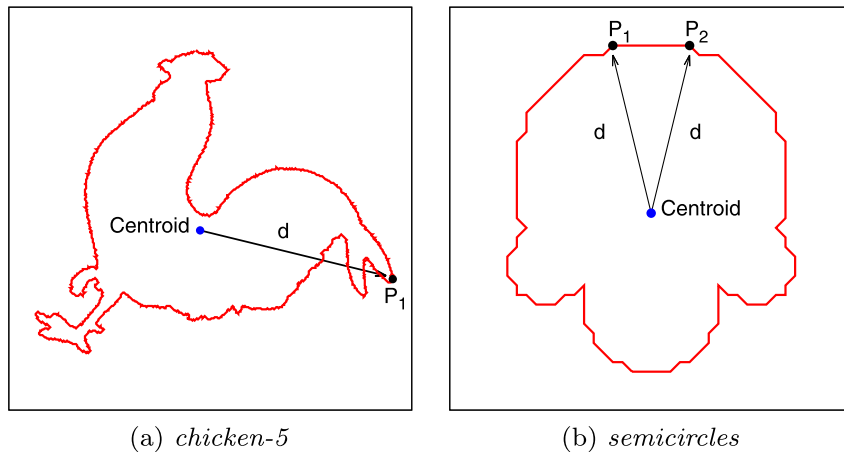


Fig. 6. Third step – Proposed method for the computation of the significance of the initial points: d is the maximum distance from the centroid to the contour points.

most perceptually significant straight line segments. In order to use the RDP method to generate the polygonal approximation of a closed contour, the following steps can be applied:

1. Find the top-left and bottom-right points of the contour. These points are considered as *initial dominant points*.
2. The farthest point from the line segment defined by two consecutive dominant points is found [8]. The selected point is considered as new dominant point.
3. Repeat step 2 until the set of dominant points generates a polygonal approximation which fits the contour with a preset error.

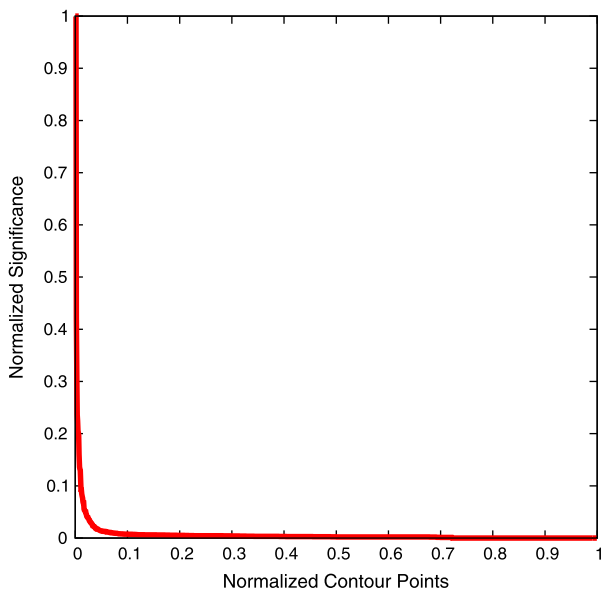


Fig. 7. Fourth step – The normalized significance curve: the plot shows the normalized significance curve generated by the contour *chicken-5* ($N = 1364$ points).

The RDP method shows two drawbacks (see Fig. 1): (1) it is a parametric method because the number of final dominant points (DP) depends on a preset error and (2) the symmetry of the contour is not taken into account. In order to avoid these drawbacks, the present work proposes a new automatic and non-parametric algorithm based on a modification of the RDP method which takes into account the symmetry of the contour (see Section 3 and Fig. 19(d)).

3. Proposed algorithm to generate polygonal approximations

The proposed algorithm consists of six steps (see Fig. 2):

1. *Selection of the initial points*: some special contour points are chosen to be considered *initial points*. See Section 3.1.
2. *Computation of the significance values of the non-initial points*: a new version of RDP algorithm [7,31], which takes into account

the symmetry of the contour, is used to compute the significance values of the non-initial points. This new symmetric version is described in Section 3.2.

3. *Computation of the significance values of the initial points*: the significance values of the *initial points* must be greater than the significance values of the other contour points (see Section 3.3).
4. *Generation of the normalized significance curve*: the significance curve must be normalized in order to facilitate the search for the dominant points using a thresholding method (see Section 3.4).
5. *Search for the threshold of the normalized significance values*: the normalized significance curve is used by a new *thresholding* method to search for the threshold of the normalized significance values. Four thresholding methods are proposed and described in Section 3.5. The performance of these thresholding methods has been analyzed in the first experiment (see Section 4.2) and the *Adaptive* method has obtained the best results and, therefore, this method has been chosen for this fifth step of the proposed algorithm.
6. *Thresholding of the normalized significance values to obtain the dominant points*: the normalized significance values of the contour points will be thresholded to obtain the dominant points of the contour. The points of the contour with normalized significance value equal to 1.0 or greater than the threshold will be chosen as *dominant points* (see Section 3.6).

In the best case, the computational complexity of the new algorithm is $\mathcal{O}(n \log(n))$; in the worst case, $\mathcal{O}(n^2)$ [9]. The following Sections 3.1–3.6 describe with more detail the six steps of the proposed algorithm to generate the polygonal approximation.

3.1. First step: selection of the initial points

The following method has been used to select the *initial points* of the polygonal approximation (see Fig. 3):

- The farthest point(s) to the centroid is (are) chosen as *initial point(s)*.
- Besides, the farthest point(s) to the previous one(s) will be also considered as *initial point(s)*.

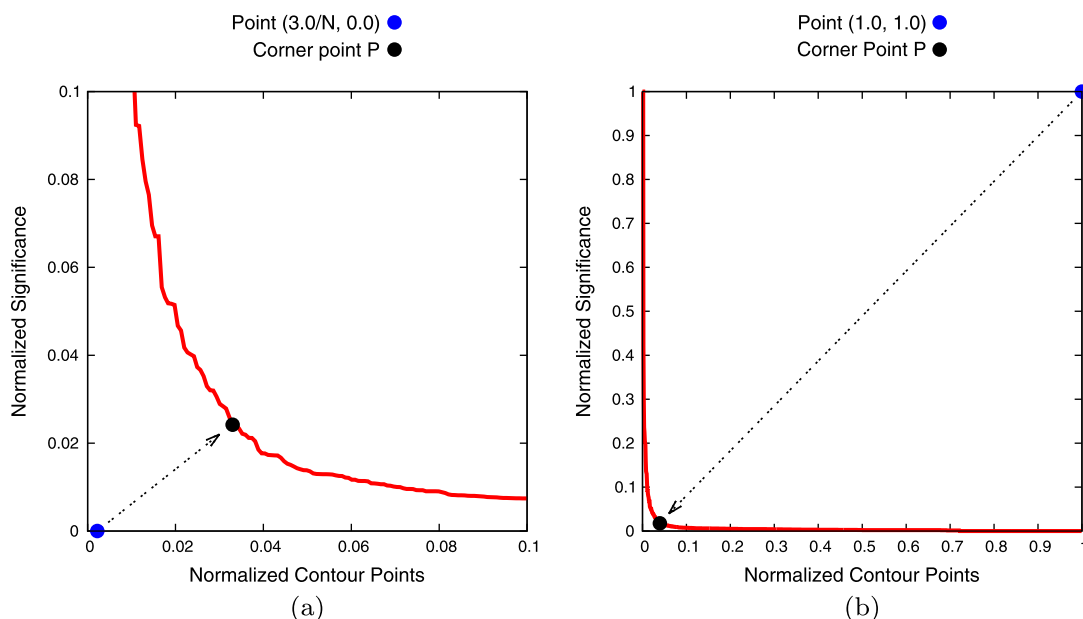


Fig. 8. Fifth step: (a) *Proximity* thresholding method: the threshold h is the ordinate y of the nearest point P to the point $(3.0/N, 0.0)$, where N is the number of contour points (zoom on $[0.0, 0.1]$). (b) *Distance* thresholding method: the threshold h is the ordinate y of the farthest point P to the point $(1.0, 1.0)$.

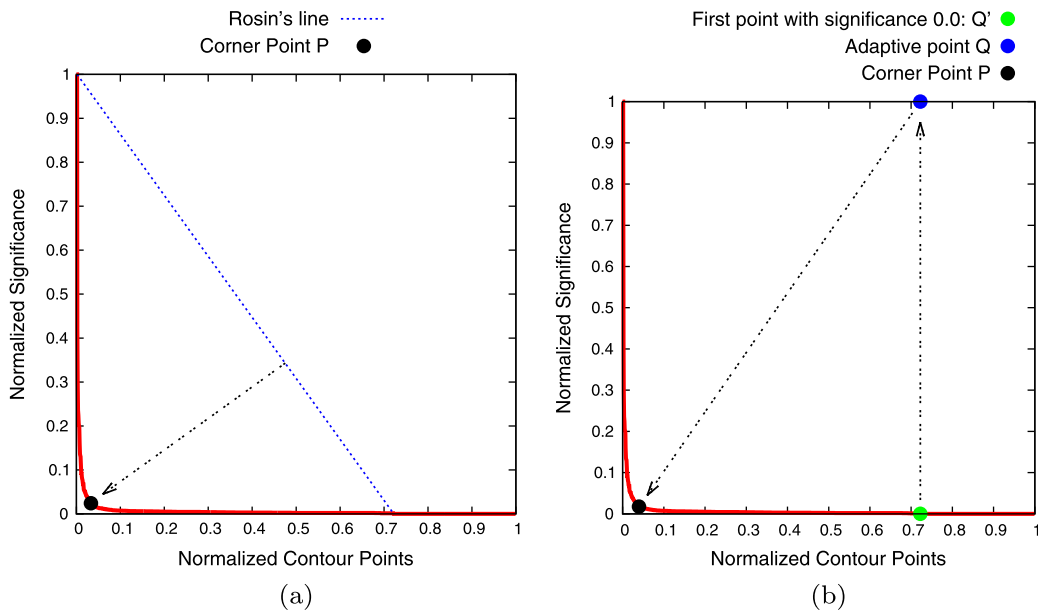


Fig. 9. Fifth step: (a) Rosin thresholding method: the threshold h is the ordinate y of the farthest point P to the Rosin's line. (b) Adaptive thresholding method: the threshold h is the ordinate y of the farthest point P to the Adaptive point Q .

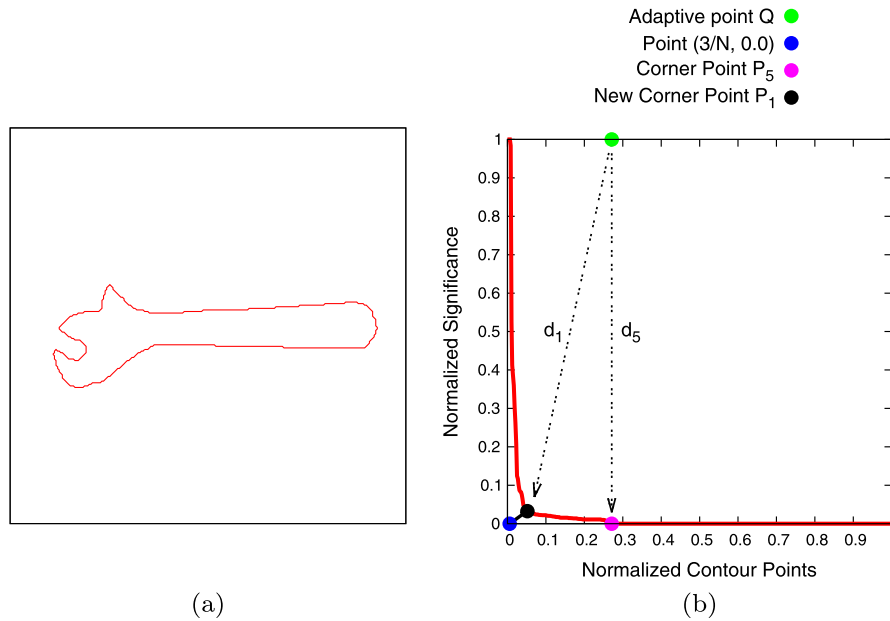


Fig. 10. Special case: (a) contour *adjustable-spanner* ($N = 484$ points; only 131 break points = 27.07%) and (b) its normalized significance curve. The corner point P_4 proposed by the fourth method is changed by the corner point P_1 proposed by the first method. ($d_1 = 0.9919, d_4 = 1.0$).

Table 1
First experiment: performance comparison of the new algorithm using different thresholding methods.

Thresholding method	Average values							
	DP	CR	E_∞	ISE	WE_∞	WE	WE_2	WE_3
Proximity	32.09	41.16	4.25	3908.34	0.122	91.812	2.428	0.073
Distance	33.18	40.97	3.98	3713.35	0.115	81.531	2.134	0.066
Rosin	25.01	56.03	6.22	9576.40	0.131	145.134	2.868	0.067
Adaptive	45.07	30.97	2.76	2024.67	0.105	53.976	1.751	0.069

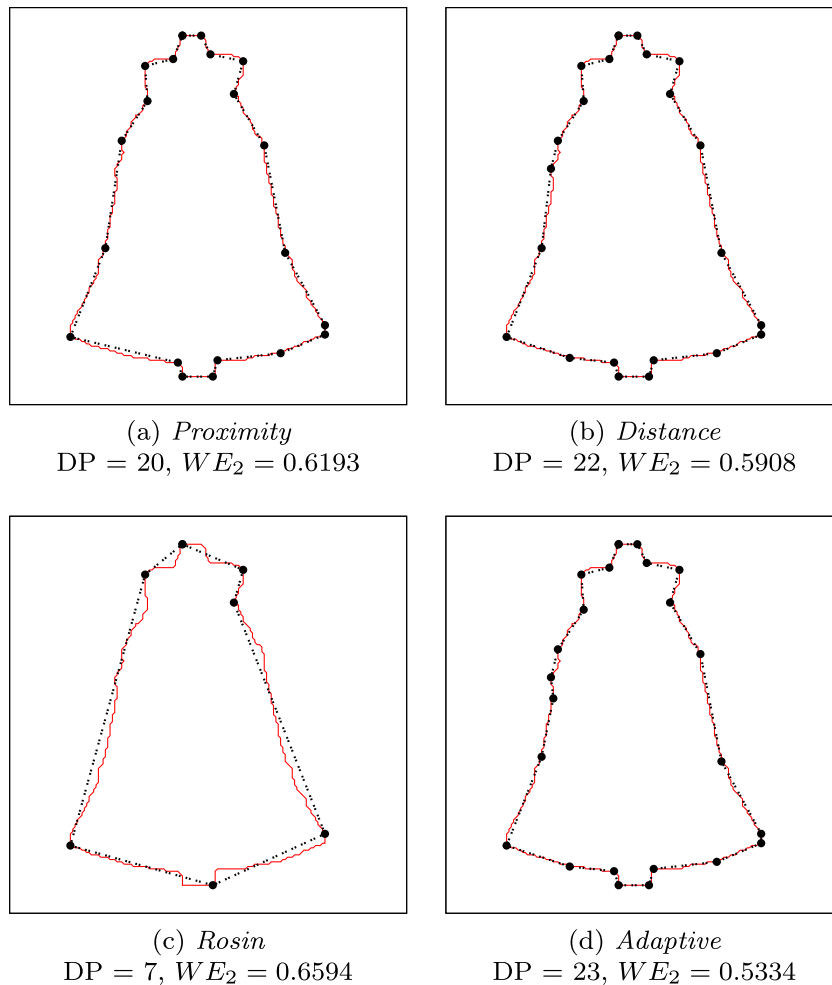
Note: the average number of contour points of MPEG-7 CE-Shape-1 (Part B) is 1271.04 points.

Table 2

Second experiment: performance comparison of the algorithms.

Algorithm	Average values							
	DP	CR	E_∞	ISE	WE_∞	WE	WE_2	WE_3
RDP	110.12	17.78	1.23	219.34	0.103	19.554	2.266	0.313
Carmona	67.70	32.59	3.63	2168.16	0.159	61.959	3.164	0.242
Masood	132.15	39.52	5.10	91376.80	0.114	279.230	3.451	0.505
Proposal	45.07	30.97	2.77	2024.67	0.105	53.976	1.751	0.069

Note: the average number of contour points of MPEG-7 CE-Shape-1 (Part B) is 1271.04 points.

**Fig. 11.** First experiment: polygonal approximations for *bell-7* ($N = 406$ points) using the proposed thresholding methods.

This method is independent of the starting point, invariant to rotations and scales and takes into account the symmetry of the contour. The computational complexity of this step is $\mathcal{O}(n)$.

3.2. Second step: computation of the significance of the non-initial points

The proposed method to compute the significance of the *non-initial* points modifies the original RDP method [7,31] to take into account the symmetry of the contour. The new version consists of the following steps:

1. Let $I = \{P_1, \dots, P_n\}$ be the set of *initial points* obtained by the method described in the first step (see Section 3.1):

2. Run the recursive procedure: *compute_significance_values*

- The following recursive procedure

compute_significance_values(P, P')

is run for every two consecutive initial points P_i and $P_{(i+1) \bmod n}$ ($\forall i \in \{1, \dots, n\}$):

- Compute the maximum distance d from the contour points located between P and P' to the segment defined by P and P' .
- If d is equal to 0.0, then assign the significance value 0.0 to every contour point located between P and P' . In this case, the recursive process will stop.
- On the contrary, if d is not equal to 0.0, then

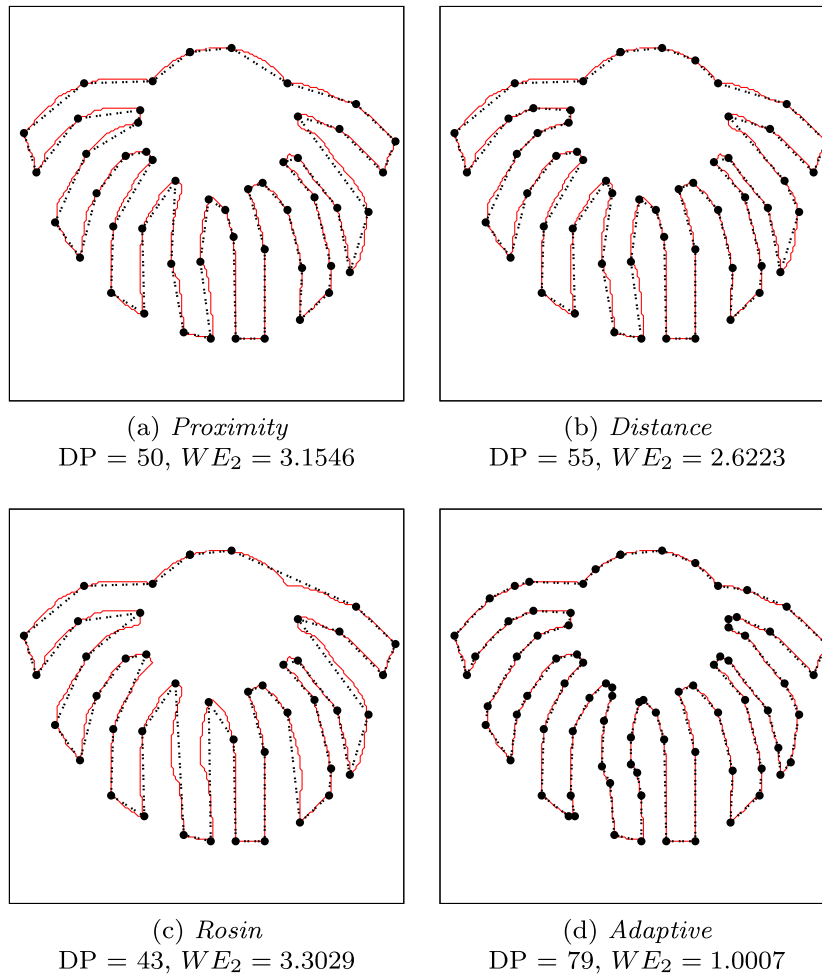


Fig. 12. First experiment: polygonal approximations for *octopus-14* ($N = 1211$ points) using the proposed thresholding methods.

- i. Select the farthest contour point(s) to the segment defined by P and P' .
 - In most cases, there will be only a single point, and the procedure will continue in the step ii.
 - However, there could be more than one point, for instance, if the contour has a straight side parallel to the segment defined by P and P' . In this situation, select from among them only the farthest points to the centroid of the contour, which will be considered as *candidate points* (see Fig. 5).
 - This step takes into account the symmetry of the contour.
- ii. Be Q_1, Q_2, \dots, Q_m the selected *candidate points*:
 - Assign the maximum distance d as *significance value* of every point $Q_i, i \in \{1, \dots, m\}$.
 - Run the recursive process *compute_significance_values* with every pair of points P and Q_1 , Q_1 and Q_2, \dots, Q_{m-1} and Q_m and, finally, Q_m and P' .

In summary, the significance value of every *non-initial* point is its deviation error d (see Fig. 4) and takes into account the symmetry of the contour. The computational complexity of this step is $\mathcal{O}(n \log(n))$ in the best case, and $\mathcal{O}(n^2)$ in the worst case [9].

3.3. Third step: computation of the significance value of the initial points

The aim of this step is to assure that the initial points will have the greatest significance value. Initially, the *maximum* of the significance value of the *non-initial* points is computed: Sig_{max} .

- If Sig_{max} is equal to 0.0, then the significance value of the *initial* points will be 1.0. This situation occurs with artificial contours in which the initial points define an exact polygonal approximation.
- On the contrary, if Sig_{max} is not equal to 0.0, then the significance value of the *initial* points will be the maximum between Sig_{max} and d , where d is the maximum distance from the centroid to the contour points (see Fig. 6).

The computational complexity of this step is $\mathcal{O}(n)$, because the maximum significance value of the contour points must be computed. The maximum distance of the contour points to the centroid was computed in the first step.

3.4. Fourth step: the normalized significance curve

The *normalized significance curve* is the resulting plot of the polygonal approximation error, defined as a function of the number of dominant points (see Fig. 7). This curve is strictly downward with a very pronounced or steep slope. At the beginning, the error reduction is significant; but, later, the error decreases when the small details of the contours are approximated. The normalization of the significance curve attempts to equate the influence of the number of points and the values of significance. The normalization consists of three steps:

1. The contour points are sorted in descending order according to their significance values.

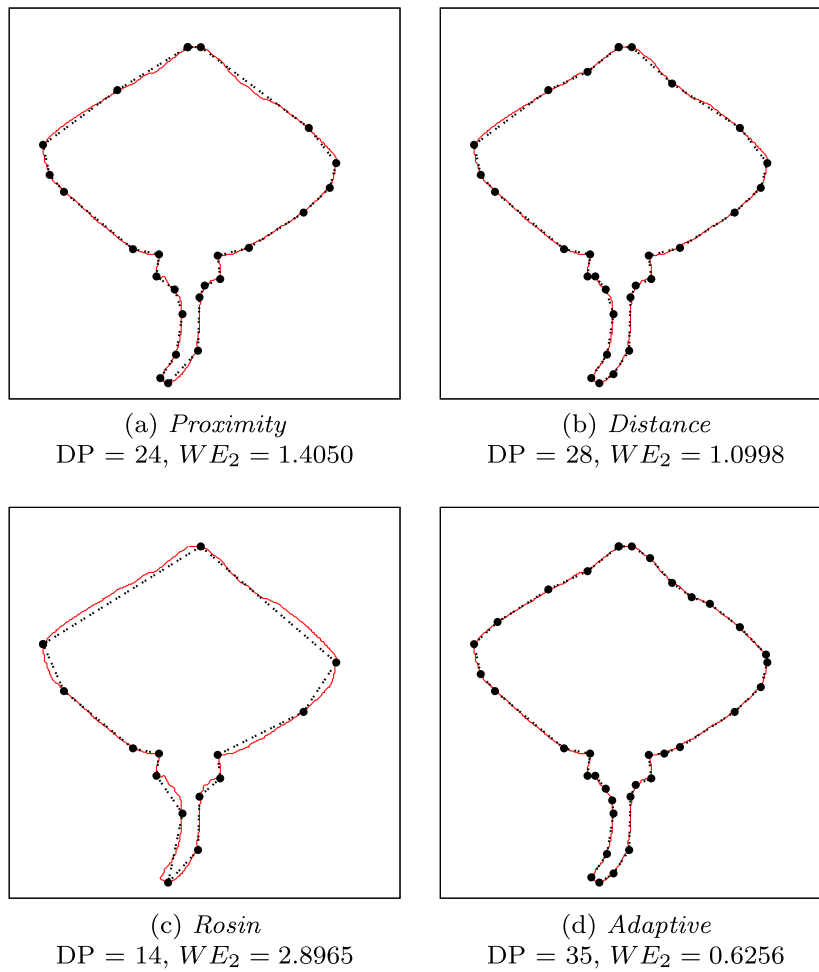


Fig. 13. First experiment: polygonal approximations for *ray-17* ($N = 689$ points) using the proposed thresholding methods.

2. The significance values are normalized in the range $[0.0, 1.0]$ dividing by the maximum significance value.
3. The number of contour points are normalized in the range $[0.0, 1.0]$ dividing by the number N of the contour points.

This normalized significance curve is used in the fifth step (see Section 3.5) to locate the *corner point*, when the behavior of the curve changes from the approximation to the important parts of the contour to the approximation to the small details.

The computational complexity of this step is $\mathcal{O}(n \log(n))$, because the normalization is a linear process $\mathcal{O}(n)$ and the normalized significance value can be sorted using an algorithm with computational complexity $\mathcal{O}(n \log(n))$.

3.5. Fifth step: search for the threshold of the normalized significance values

The aim of this fifth step is to locate the *corner point* of the normalized significance curve. This *corner point* attempts to equate the influence of the number of points of the final polygonal approximation and its deviation error. The ordinate y of the *corner point* P is the chosen value h to threshold the normalized significance values of the contour points (see Section 3.6).

The *corner point* is searched within a *region of interest*, which is defined by the interval $[Left\ extreme\ point, Right\ extreme\ point]$ of the abscissa axis, where the *Left extreme point* is the greatest normalized point with normalized significance value equal to 1.0,

and the *Right extreme point* is the smallest normalized point with normalized value of significance equal to 0.0.

Four thresholding methods have been proposed for searching the *corner point*:

1. *Proximity* method: this method searches for the point $P(x, y)$ of the normalized significance curve with the *smallest* distance to the point¹ $(3.0/N, 0.0)$, where N is the number of contour points (see Fig. 8(a)).
2. *Distance* method: this method searches for the point $P(x, y)$ of the normalized significance curve with the *greatest* distance to the point $(1.0, 1.0)$ (see Fig. 8(b)).
3. *Rosin* method: this method has been inspired by the Rosin method for unimodal thresholding [33] and searches for the point $P(x, y)$ in the region of interest of the significance curve with the *greatest* distance to the straight line defined by the points $(left\ extreme, 1.0)$ and $(right\ extreme, 0.0)$. This straight line is called *Rosin's line* (see Fig. 9(a)).
4. *Adaptive* method: this method searches for the *farthest* point $P(x, y)$, in the region of interest of the significance curve, to the *Adaptive point* $Q(x_0, 1.0)$, where the project point $Q'(x_0, 0.0)$ is the first point in ascending order with normalized significance value equal to 0.0 (see Fig. 9(b)).

In a few special cases, the *Adaptive* thresholding method does not work properly when the number of break points is much less than the number of contour points. In such a case, the distance

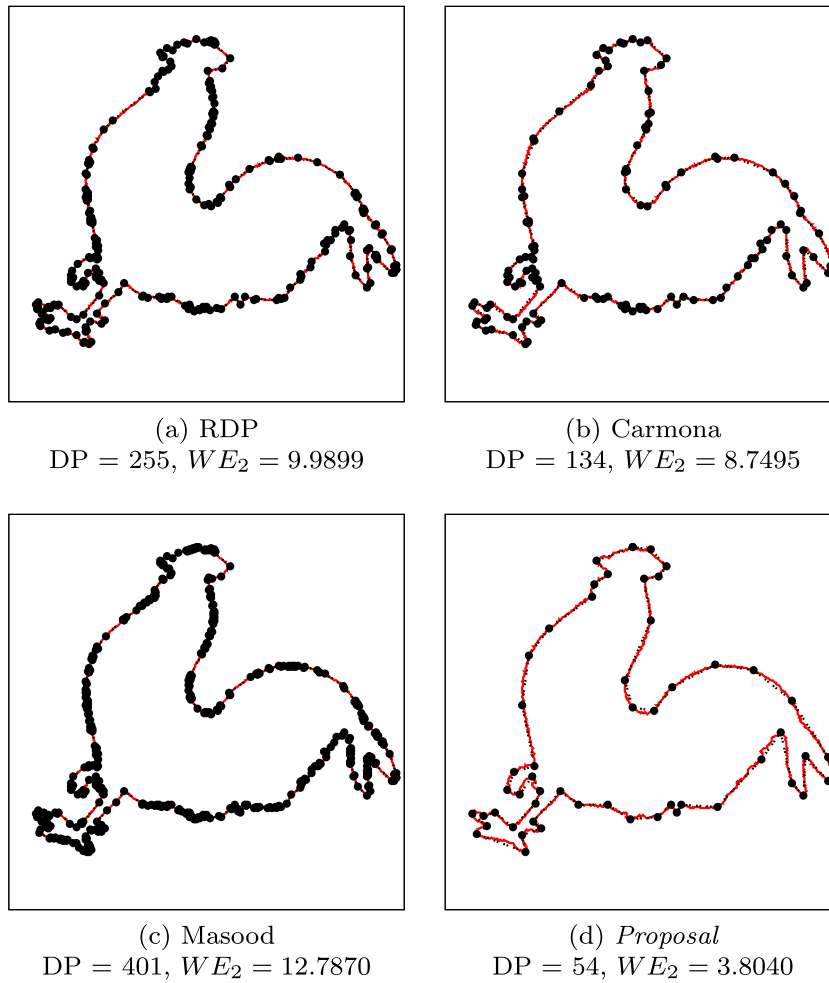


Fig. 14. Second experiment: polygonal approximations for *chicken-5* ($N = 1364$ points).

from the *Adaptive point* $Q(x_0, 1.0)$ to the *farthest point* $P(x, y)$ is equal to 1.0 and greater than the distance from the *Adaptive point* to the *corner point* (see Fig. 10(b)). The threshold h proposed by the *Adaptive* method is 0.0 and all the break points would be considered as dominant points, which would be undesirable. In order to avoid this drawback, if the threshold h proposed by the *Adaptive* method is 0.0, then the threshold h proposed by the *Proximity* method will be used instead.

The performances of these four thresholding methods have been analyzed empirically (see the first experiment in Section 4.2) and the *Adaptive* method, with the modification for special cases described previously, has obtained the best results and, therefore, it has been chosen for the fifth step of the new algorithm.

The computational complexity of this fifth step is linear: $\mathcal{O}(n)$.

3.6. Sixth step: thresholding of the normalized significance values

In order to obtain the *dominant points*, a thresholding method is applied to the *normalized significance values* of the contour points. If $NSig(P)$ represents the *normalized significance value* of the point P and h is the threshold proposed by any of the thresholding methods (see Section 3.5) then the *dominant points* of the contour will be the points with *normalized significance value* greater than the threshold h :

$$Dominant_Point(P) = \begin{cases} True & \text{if } (NSig(P) > h) \\ False & \text{On the contrary.} \end{cases} \quad (1)$$

The computational complexity of this step is linear: $\mathcal{O}(n)$.

4. Experiments and results

Two experiments have been developed to measure the quality of the new algorithm for dominant points detection:

- In the first experiment (see Section 4.2), the performances of the four proposed thresholding methods (see Section 3.5) have been analyzed and the *Adaptive* thresholding method has obtained the best results. This method has been chosen for the fifth step of the new algorithm for polygonal approximations.
- In the second experiment (see Section 4.3), the new proposal, using the *Adaptive* thresholding method, has been compared with the non-parametric dominant point detection algorithms proposed by Prasad et al. [30].

The database of shapes “MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)” [12] has been used in the experiments [26,27]. This database is available in [16] and contains 1400 images, classified into 70 categories, and each category includes 20 samples, with different rotation, size and position, and even image resolution [15]. The shapes contours have been extracted using the algorithm implemented in OpenCV 2.4.6. The contours obtained from the shapes of the database are affected by digitization and noise problems that occur in real situations.

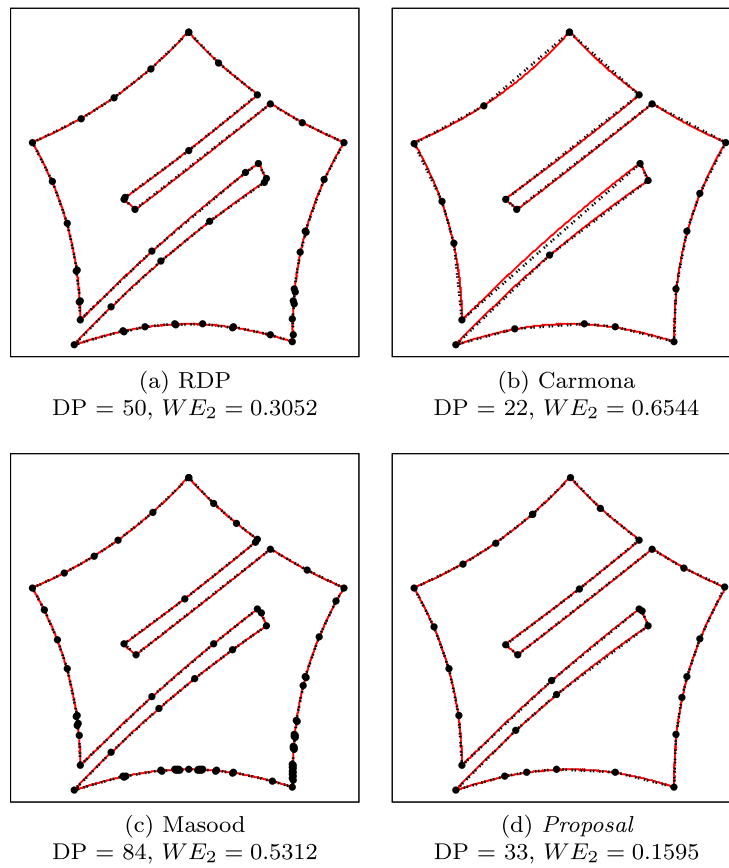


Fig. 15. Second experiment: polygonal approximations for *device6-9* ($N = 1590$ points).

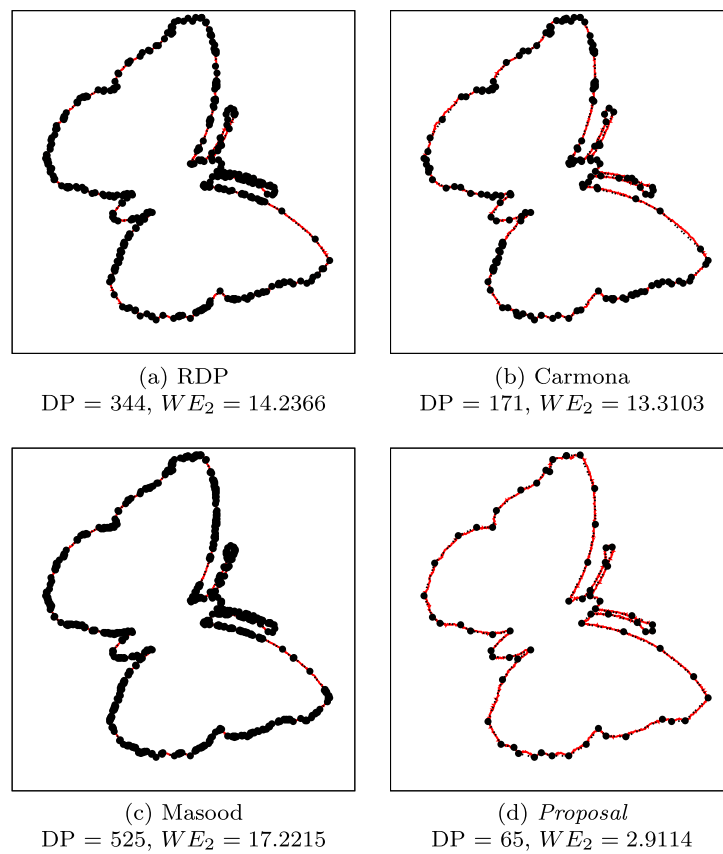


Fig. 16. Second experiment: polygonal approximations for *butterfly-13* ($N = 1786$ points).

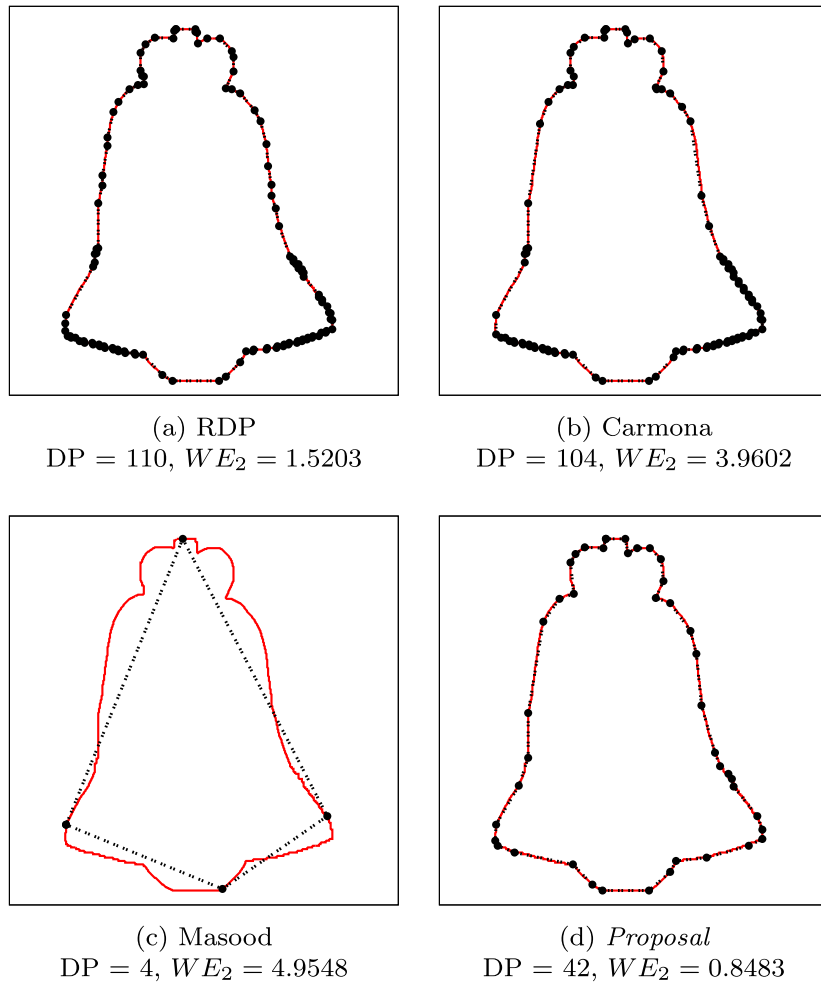


Fig. 17. Second experiment: polygonal approximations for bell-10 ($N = 1202$ points).

4.1. Performance quality measures

In order to evaluate the quality of the polygonal approximations generated by the algorithms, several measures have been proposed [6,14,26,27,32]. *Compression ratio* $CR = \frac{n}{n_d}$, where n is the number of contour points and n_d is the number of dominant points, measures the normalized compression rate of the polygonal approximation. On the other hand, the *fitting* of the polygonal approximation to the original contour can be evaluated using several measures: (a) *Integral square error* $ISE = \sum_{i=1}^n e_i^2$, where e_i is the distance of contour point P_i to the polygonal approximation; (b) *Maximum error* $E_\infty = \max_{i \in \{1, \dots, n\}} \{e_i\}$. ISE may not take into account some relevant information of the contour when hides large deviation at particular point due to closeness of approximating polygon at other parts of the curve. Such approximation behavior is not desirable. E_∞ is proposed to solve this lack of accuracy of ISE [21].

In order to combine the *number of vertices* and the *fitting* of the polygonal approximation, some measures have been proposed. Sarkar [36] proposed the *Figure of Merit* $FOM = \frac{CR}{ISE}$ to make the tradeoff between the compression ratio (CR) and the total distortion (ISE) caused [22]. Rosin [32] showed that the two terms CR and ISE used by FOM are not balanced, causing the measure to be biased towards approximations with lower ISE (which can be easily attained by increasing the number of detected dominant points). This drawback becomes more evident for real contours, which usually contain large numbers of points. Hence, FOM is not the best measure for comparing contours with different numbers of domi-

nant points. The *weighted sum of squared error* $WE = \frac{ISE}{CR}$ is defined as the inverse of FOM [38,39]. The *weighted maximum error* $WE_\infty = \frac{E_\infty}{CR}$ has also been proposed [18,21,39]. Technically, WE and WE_∞ are similar to FOM and suffer similar problems [21].

A parameterized version $WE_n = \frac{ISE}{(CR)^n}$ has also been proposed [5,20,25,26] to balance the contribution of ISE and CR , where $n = 1, 2, 3$. Carmona [4] demonstrated that the value $n = 2$ obtains the best performance.

Rosin [32] proposed a new measure to avoid the drawback of FOM : $Merit = \sqrt{Fidelity \times Efficiency}$, where $Fidelity = \frac{E_{opt}}{E_{appr}} \times 100$ and $Efficiency = \frac{N_{opt}}{N_{appr}} \times 100$ where E_{appr} and N_{appr} are the error and the number of dominant points of suboptimal polygonal approximation, E_{opt} is the error produced by the optimal algorithm with the same number of dominant points and N_{opt} represents the number of dominant points that would require an optimal algorithm to produce the same error [28]. The *Fidelity* measures how well the polygon obtained by the algorithm to be tested fits the curve relative to the optimum polygon, in terms of the approximation error. The *Efficiency* measures the compactness of the polygon obtained by the algorithm to be tested, relative to the optimum polygon that incurs the same error [32].

The Rosin's evaluation can compare results of different algorithms with different number of dominant points. Nevertheless, Masood [22] pointed out that this method also suffers a few weaknesses. The polygon that consists of just *break points* will produce $Fidelity = 100$, $Efficiency = 100$ and $Merit = 100$. It means that the set

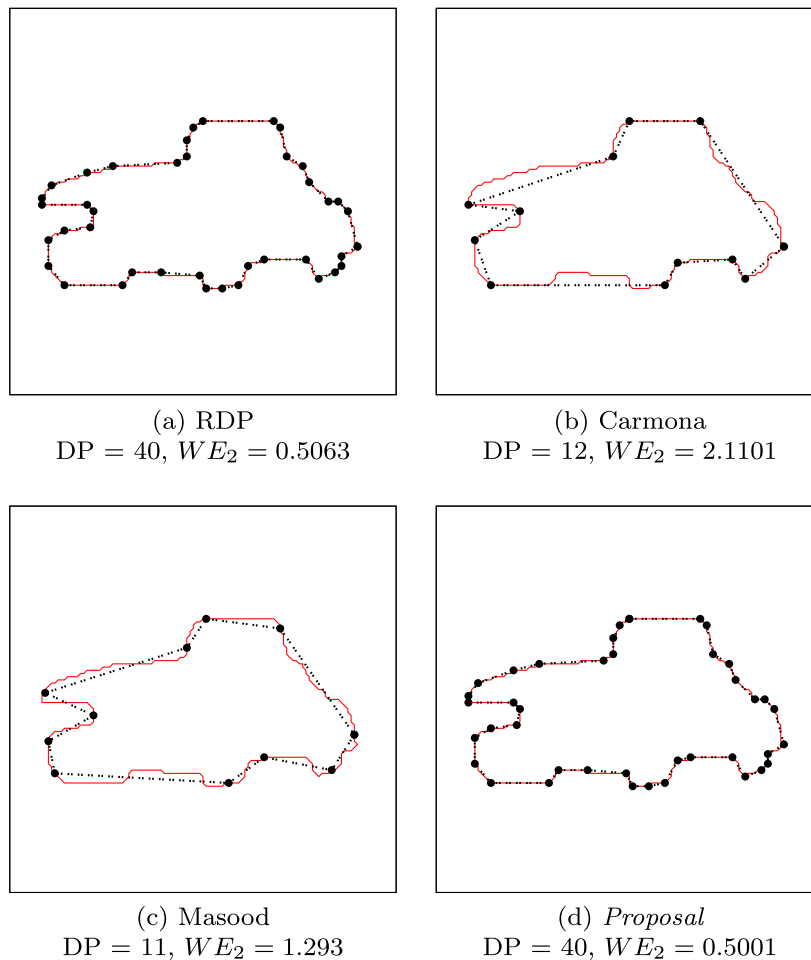


Fig. 18. Second experiment: polygonal approximations for truck-07 ($N = 277$ points).

of break points taken as dominant points will produce a perfect approximation, whereas this type of approximation is of no practical use since its compression ratio is very low. Carmona [6] proposed a new measure for assessing polygonal approximation of curves. This measure uses the method of Pérez and Vidal [28] and the optimization of an objective function based on the WE_2 . The main drawback of these measures proposed by Rosin [32] and Carmona [6] is that they need optimal solutions which are computationally very expensive [28].

In summary, when the number of dominant points (DP) is the same, ISE and E_∞ are the most appropriate quantitative measures for comparison of polygonal approximations [21]. If the number of DP is different, then the parameterized WE_2 has showed better performance [4]. In the experiments, for the sake of completeness, the values of DP , CR , ISE , E_∞ , WE , WE_2 and WE_3 have been used (see Tables 1 and 2).

4.2. First experiment: selection of the best thresholding method of the fifth step of the new algorithm

The new algorithm for polygonal approximations consists of six steps (see Section 3), but four *thresholding* methods have been proposed for the fifth step (see Section 3.5): *Proximity*, *Distance*, *Rosin* and *Adaptive* methods. The aim of this first experiment is to select the best thresholding method.

Taking into account the proposed thresholding methods, four versions of the new algorithm have been applied to 1400 contours extracted from the shapes of the MPEG-7 Core Experiment

CE-Shape-1 Test Set (Part B) database [12]. Figs. 11–13 show some polygonal approximations generated using the four thresholding methods. These figures show that the *Adaptive* thresholding method generates the best polygonal approximations. In order to facilitate the comparisons, the figures indicate the number of dominant points (DP) and the value of WE_2 .

The quantitative evaluation of the thresholding methods is based on Table 1, that shows the average values of the quality measures. The *Adaptive* method has obtained the best results with the measures WE_∞ , WE , and WE_2 and a very good result, in relation to the best, with the measure WE_3 . The *Adaptive* method generates polygonal approximations with a few more dominant points (DP) than the other methods, but the fitting to the contour is significantly improved (see Figs. 11–13). Therefore, the *Adaptive* method has been chosen as the *thresholding* method of the fifth step of the new algorithm. Once the best thresholding method has been chosen, the new algorithm can *automatically* generate polygonal approximations of closed digital planar curves without using any parameters.

4.3. Second experiment: comparison with the non-parametric algorithms proposed by Prasad

In the second experiment, the new proposal for polygonal approximations, using the *Adaptive* thresholding method, has been compared with the versions of RDP, Carmona and Masood algorithms proposed by Prasad et al. [30]. These algorithms have been

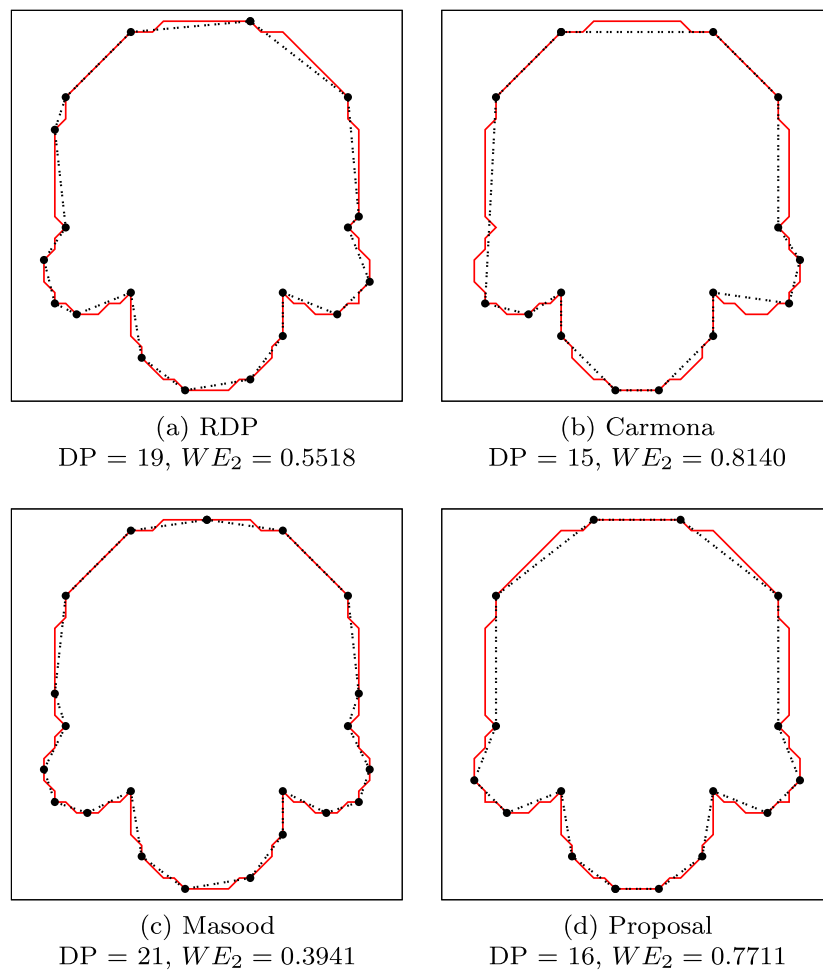


Fig. 19. Second experiment: polygonal approximations for *semicircles* ($N = 102$ points). Only the new *proposal* generates a symmetric polygonal approximation.

selected because they can automatically generate polygonal approximations of digital planar curves.

The computational complexity of the proposed algorithm is $\mathcal{O}(n \log(n))$, which is similar to the computational complexity of the RDP method [9]. The computational complexity of the Carmona algorithm is $\mathcal{O}((m - n_d)n)$, where n is the number of the contour points, m is the number of initial break points, and n_d is the number of final dominant points [5]. The computational complexity of the Masood method can be given as $\mathcal{O}((m - n_d)n^2)$ owing to its optimization process [23].

The algorithms have been applied to 1400 contours extracted from the shapes of MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B) database [12] to analyze their performances. Figs. 14–18 show some polygonal approximations generated by these algorithms. Besides, the well-known contour *semicircles* has been used to show the performances of the algorithms with symmetric contours (see Fig. 19). In order to facilitate the comparisons, the figures indicate the number of dominant points (DP), and the value of WE_2 .

Figs. 14(d)–18(d) show that the polygonal approximations generated by the new proposal properly fit to the contours. On the contrary, the performance of the algorithms proposed by Prasad et al. depends on the contours. For instance, these algorithms generate many dominant points for some contours (see Figs. 14, 16 and 17) and very few dominant points for other contours (see Figs. 17(c) and 18(b) and (c)).

The new proposal is not affected by the digitization noise of the contours used in the experiments (see, for instance, Fig. 15). In addition, the new algorithm takes into account the symmetry of

the contours, whereas the versions proposed by Prasad et al. do not. See the polygonal approximations generated for the contour *semicircles* in Fig. 19.

Table 2 shows the average values of the quality measures obtained by the algorithms. The performance of the algorithms depends on the quality measures. The new proposal has obtained the best results for WE_2 and WE_3 , and the second best results for WE_∞ and WE . This table also shows that the new method generates far fewer dominant points (DP) than the other methods. In our opinion, this experiment shows, once again, that WE (inverse of FOM) is not the best measure for comparing polygonal approximations with different numbers of dominant points because it is biased towards approximations with lower ISE , what it can be easily achieved by increasing the number of dominant points [20].

In summary, the second experiment has shown that the new proposal has good performance for generating polygonal approximations of 2D closed contours, and it has outperformed the non-parametric versions proposed by Prasad et al. [30].

5. Conclusions

The present paper has proposed a new approach for polygonal approximations of closed digital planar curves. The new proposal consists of six steps and is focused on detecting dominant points which are usually used for automatically generating a polygonal approximation of the given closed curves. The significance values of the contour points are computed using a new version of the RDP method taking into account the *symmetry* of the contour,

which is a common feature of the natural shapes. The final dominant points are located using a new *Adaptive* method to threshold the normalized significance values of the contour points. The proposed algorithm is *automatic* and *non-parametric*, and its computational complexity is $\mathcal{O}(n \log(n))$. The experiments have shown that the new algorithm can generate efficient and effective polygonal approximations for digital planar curves.

Finally, the proposed approach based on a *thresholding* method can be applied by any dominant points detection algorithm that assigns a significance values to the contour points.

References

- [1] E.J. Aguilera-Aguilera, A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Muñoz-Salinas, Novel method to obtain the optimal polygonal approximation of digital planar curves based on Mixed Integer Programming, *J. Vis. Commun. Image R.* 30 (2015) 106–116, <http://dx.doi.org/10.1016/j.jvcir.2015.03.007>.
- [2] E. Ataer-Cansizoglu, E. Bas, J. Kalpathy-Cramer, G.C. Sharp, D. Erdogmus, Contour-based shape representation using principal curves, *Patt. Recog.* 46 (2013) 1140–1150, <http://dx.doi.org/10.1016/j.patcog.2012.10.014>.
- [3] F. Attneave, Some informational aspects of visual perception, *Psychol. Rev.* 61 (1954) 189–193.
- [4] A. Carmona-Poyato, N.L. Fernández-García, R. Medina-Carnicer, F.J. Madrid-Cuevas, Dominant point detection: a new proposal, *Image Vis. Comput.* 23 (2005) 1226–1236, <http://dx.doi.org/10.1016/j.imavis.2005.07.025>.
- [5] A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Medina-Carnicer, R. Muñoz-Salinas, Polygonal approximation of digital planar curves through break point suppression, *Patt. Recog.* 43 (2010) 14–25, <http://dx.doi.org/10.1016/j.patcog.2009.06.010>.
- [6] A. Carmona-Poyato, R. Medina-Carnicer, F.J. Madrid-Cuevas, R. Muñoz-Salinas, N.L. Fernández-García, A new measurement for assessing polygonal approximation of curves, *Patt. Recog. Lett.* 4 (2011) 44–54, <http://dx.doi.org/10.1016/j.patcog.2010.07.029>.
- [7] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *Cartograph. Int. J. Geograph. Inf. Geovisual.* 10 (2) (1973) 112–122, <http://dx.doi.org/10.3138/FM57-6770-U75U-7727>.
- [8] K. Ebisch, A correction to the Douglas-Peucker line generalization algorithm, *Comp. Geosci.* 28 (2002) 995–997, [http://dx.doi.org/10.1016/S0098-3004\(02\)00009-2](http://dx.doi.org/10.1016/S0098-3004(02)00009-2).
- [9] J. Hershberger, J. Snoeyink, Speeding up the Douglas-Peucker line simplification algorithm, in: P. Bresnahan (Ed.), *Proceedings of the Fifth International Symposium on Spatial Data Handling*, Charleston SC, 1992, pp. 134–143.
- [10] J.-H. Horn, J.T. Li, An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, *Patt. Recog. Lett.* 23 (2002) 171–182, [http://dx.doi.org/10.1016/S0167-8655\(01\)00098-8](http://dx.doi.org/10.1016/S0167-8655(01)00098-8).
- [11] J.M. Iñesta, M. Buendía, M.A. Sart, Reliable polygonal approximations of imaged real objects through dominant point detection, *Patt. Recog.* 31 (1998) 685–697, [http://dx.doi.org/10.1016/S0031-3203\(97\)00081-2](http://dx.doi.org/10.1016/S0031-3203(97)00081-2).
- [12] S. Jeannin, M. Bober, Description of Core Experiments for MPEG-7 Motion/Shape, MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG99/N2690, Seoul, March 1999.
- [13] A. Kolesnikov, P. Fränti, Polygonal approximation of closed discrete curves, *Patt. Recog.* 40 (2007) 1282–1293, <http://dx.doi.org/10.1016/j.patcog.2006.09.002>.
- [14] A. Kolesnikov, t. Kauranne, Unsupervised segmentation and approximation of digital curves with rate-distortion curve modeling, *Patt. Recog.* 47 (2014) 623–633, <http://dx.doi.org/10.1016/j.patcog.2013.09.002>.
- [15] L.J. Latecki, R. Lakämper, Ulrich Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000, pp. 424–429.
- [16] MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B), Benchmarking Image Database for Shape Recognition Techniques <<http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz>>.
- [17] S. Loncaric, A survey of shape analysis techniques, *Patt. Recog.* 31 (1998) 983–1001, [http://dx.doi.org/10.1016/S0031-3203\(97\)00122-2](http://dx.doi.org/10.1016/S0031-3203(97)00122-2).
- [18] D.G. Lowe, Three dimensional object recognition from single two dimensional images, *Artif. Intell.* 31 (1987) 355–395, [http://dx.doi.org/10.1016/0004-3702\(87\)90070-1](http://dx.doi.org/10.1016/0004-3702(87)90070-1).
- [19] M. Marji, P. Siy, A new algorithm for dominant point detection and polygonization of digital curves, *Patt. Recog.* 36 (2003) 2239–2251, [http://dx.doi.org/10.1016/S0031-3203\(03\)00119-5](http://dx.doi.org/10.1016/S0031-3203(03)00119-5).
- [20] M. Marji, P. Siy, Polygonal representation of digital planar curves through dominant point detection—a nonparametric algorithm, *Patt. Recog.* 37 (2004) 2113–2130, <http://dx.doi.org/10.1016/j.patcog.2004.03.004>.
- [21] A. Masood, S.A. Haq, A novel approach to polygonal approximation of digital curves, *J. Vis. Commun. Image Represent.* 18 (2007) 264–274, <http://dx.doi.org/10.1016/j.jvcir.2006.12.002>.
- [22] A. Masood, Dominant point detection by reverse polygonization of digital curves, *Image Vis. Comput.* 26 (2008) 702–715, <http://dx.doi.org/10.1016/j.imavis.2007.08.006>.
- [23] A. Masood, Optimized polygonal approximation by dominant point deletion, *Patt. Recog.* 41 (2008) 227–239, <http://dx.doi.org/10.1016/j.patcog.2007.05.021>.
- [24] A. Melkman, J. O'Rourke, On polygonal chain approximation, in: G.T. Toussaint (Ed.), *Computational Morphology*, North-Holland, Amsterdam, 1998, pp. 87–95.
- [25] T.P. Nguyen, I. Debled-Rennesson, A discrete geometry approach for dominant point detection, *Patt. Recog.* 44 (1) (2011) 32–44, <http://dx.doi.org/10.1016/j.patcog.2010.06.022>.
- [26] M.T. Parvez, S.A. Mahmoud, Polygonal approximation of digital planar curves through adaptive optimizations, *Patt. Recog. Lett.* 31 (2010) 1997–2005, <http://dx.doi.org/10.1016/j.patrec.2010.06.007>.
- [27] M.T. Parvez, Optimized polygonal approximations through vertex relocations in contour neighborhoods, *Image Vis. Comput.* 34 (2015) 1–10, <http://dx.doi.org/10.1016/j.imavis.2014.10.012>.
- [28] J.C. Perez, E. Vidal, Optimum polygonal approximation of digitized curves, *Patt. Recog. Lett.* 15 (1994) 743–750, [http://dx.doi.org/10.1016/0167-8655\(94\)90002-7](http://dx.doi.org/10.1016/0167-8655(94)90002-7).
- [29] A. Pikaz, I. Dinstein, Optimal polygonal approximation of digital curves, *Patt. Recog.* 28 (1995) 373–379, [http://dx.doi.org/10.1016/0031-3203\(94\)00108-X](http://dx.doi.org/10.1016/0031-3203(94)00108-X).
- [30] D.K. Prasad, M.K.H. Leung, C. Quek, S.-Y. Cho, A novel framework for making dominant point detection methods non-parametric, *Image Vis. Comput.* 30 (2012) 843–859, <http://dx.doi.org/10.1016/j.imavis.2012.06.010>.
- [31] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, *Comput. Graph. Image Process.* 1 (3) (1972) 244–256, [http://dx.doi.org/10.1016/S0146-664X\(72\)80017-0](http://dx.doi.org/10.1016/S0146-664X(72)80017-0).
- [32] P.L. Rosin, Techniques for assessing polygonal approximations of curves, *IEEE Trans. Patt. Anal. Mach. Intell.* 19 (1997) 659–666, <http://dx.doi.org/10.1109/34.601253>.
- [33] P.L. Rosin, Unimodal thresholding, *Patt. Recog.* 34 (2001) 2083–2096, [http://dx.doi.org/10.1016/S0031-3203\(00\)00136-9](http://dx.doi.org/10.1016/S0031-3203(00)00136-9).
- [34] M. Salotti, An efficient algorithm for the optimal polygonal approximation of digitized curves, *Patt. Recog. Lett.* 22 (2001) 215–221, [http://dx.doi.org/10.1016/S0167-8655\(00\)00088-X](http://dx.doi.org/10.1016/S0167-8655(00)00088-X).
- [35] M. Salotti, Optimal polygonal approximation of digitized curves using the sum of square deviations criterion, *Patt. Recog.* 35 (2002) 435–443, [http://dx.doi.org/10.1016/S0031-3203\(01\)00051-6](http://dx.doi.org/10.1016/S0031-3203(01)00051-6).
- [36] D. Sarkar, A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves, *Patt. Recog.* 14 (1993) 959–964, [http://dx.doi.org/10.1016/0167-8655\(93\)90004-W](http://dx.doi.org/10.1016/0167-8655(93)90004-W).
- [37] C. Urdiales, A. Bandera, F. Sandoval, Non-parametric planar shape representation based on adaptive curvature functions, *Patt. Recog.* 35 (2002) 43–53, [http://dx.doi.org/10.1016/S0031-3203\(01\)00041-3](http://dx.doi.org/10.1016/S0031-3203(01)00041-3).
- [38] W.Y. Wu, Dominant point detection using bending value, *Image Vis. Comput.* 21 (2003) 517–525, [http://dx.doi.org/10.1016/S0262-8856\(03\)00031-3](http://dx.doi.org/10.1016/S0262-8856(03)00031-3).
- [39] W.Y. Wu, An adaptive method for detecting dominant points, *Patt. Recog.* 36 (2003) 2231–2237, [http://dx.doi.org/10.1016/S0031-3203\(03\)00087-6](http://dx.doi.org/10.1016/S0031-3203(03)00087-6).
- [40] D. Zhang, G. Lu, Review of shape representation and description techniques, *Patt. Recog.* 37 (2004) 1–19, <http://dx.doi.org/10.1016/j.patcog.2003.07.008>.

Bibliografía

- [1] P.-Y. Yin. *Pattern Recognition Techniques, Technology and Applications*. Editor: Peng-Yeng Yin. InTech, 2008. ISBN: 978-853-7619-24-4. DOI: 10.5772/90.
- [2] Y. Aloimonos. Visual shape computations. *Proceedings of the IEEE*, 76:899–916, 1988. doi: 10.1109/5.5964.
- [3] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982. ISBN: 0131653164.
- [4] L. S. Davis. *Handbook of Pattern Recognition and Image Processing*. Editores: Y. Young and K. S Fu. Academic Press, 1986. ISBN: 0127745610.
- [5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973. ISBN: 0471223611.
- [6] B. K. P. Horn. *Robot Vision*. The MIT Electronical Computer Science. McGraw-Hill Book Company, 1986. ISBN: 0262081598.
- [7] A. K. Jain. *Fundamentals of Digital image Processing*. Information and System Sciences series. Prentice Hall, 1989. ISBN: 0133361659.
- [8] M. D. Levine. *Vision in Man and Machine*. Mc Graw-Hill, 1985. ISBN: 0070374465.
- [9] S. Loncaric. A survey of shape analysis technique. *Pattern Recognition*, 31:983–1001, 1998. doi:10.1016/S0031-2023(97)00122-2.
- [10] R. Nevatia. *Machine Perception*. Prentice Hall, 1982. ISBN: 0135419042.
- [11] P. J. Van Otterloo. *A contour-oriented approach to shape analysis*. Prentice Hall, 1991. ISBN: 0131738402.
- [12] T. Pavlidis. A review of algorithms for shape analysis. *Computer Graphics and Image Processing*, 7:243–258, 1978. doi: 10.1016/0146-664X(78)90115-6.

- [13] T. Pavlidis. Algorithms for shape analysis of contours and waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:301–312, 1980. doi: 10.1109/T-PAMI.1980.4767029.
- [14] R. J. Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley and Sons, 1989. ISBN: 0471857181.
- [15] L. Uhr. *Pattern Recognition: Theory, Experiments, Computer Simulations and Dynamical Models of Forms, Perception and Discovery*. Wiley, 1966.
- [16] L. Young and K. S. Fu. *Handbook of Pattern Recognition and Image Processing*. Academic Press, isbn: 9780080926667 1986.
- [17] A. Masood and S.A. Haq. A novel approach to polygonal approximation of digital curves. *J. Vis. Commun. Image R*, 18:264–274, 2007.
- [18] A. Masood. Dominant point detection by reverse polygonization of digital curves. *Image and Vision Computing*, 26:702–715, 2008. doi: 10.1016/j.imavis.2007.08.006.
- [19] M. Marji. Polygonal representation of digital planar curves through dominant point detection. *Pattern Recognition*, 37, 2004. doi: 10.1016/j.patcog.2004.03.004.
- [20] P. Cornic. Another look at the dominant point detection of digital curves. *Pattern Recognition Letters*, 18:13–25, 1997. doi: 10.1016/j.patcog.2004.03.004.
- [21] Wenhua Wan and José A. Ventura. Segmentation of planar curves into straight-line segments and elliptical arcs. *Graphical Models and Image Processing*, 59(6):484–494, November 1997. doi: 10.1006/gmip.1997.0450.
- [22] Günter Rote R.L. Scot Drysdale and Astrid Sturm. Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs. *Computational Geometry*, 41:31–47, 2008. doi: 10.1016/j.comgeo.2007.10.009.
- [23] P. L. Rosin. Nonparametric segmentation of curves into various representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1140–1153, December 1995. doi: 10.1109/34.476507.
- [24] Soo-Chang Pei and Ji-Hwei Horng. Fitting digital curve using circular arcs. *Pattern Recognition*, 28(1):107–116, 1995. doi: 10.1016/0031-3203(94)00086-2.
- [25] Soo-Chang Pei and Ji-Hwei Horng. Optimum approximation of digital planar curves using circular arcs. *Pattern Recognition*, 29(3):383–388, 1996. doi: 10.1016/0031-3203(95)00104-2.
- [26] P. Ganguly S. Pal and P.K. Biswas. Cubic bézier approximation of a digitized curve. *Pattern Recognition*, 40:2730–2741, 2007. doi: 10.1016/j.patcog.2007.01.019.
- [27] Y. Nakagawa and A. Rosenfield. A note on polygonal and elliptical approximation of mechanical parts. *Pattern Recognition*, 11:133–142, 1979. doi: 10.1016/0031-3203(79)90059-1.

- [28] G. Medioni and Y. Yasumoto. Corner detection and curve representation using cubic b-splines. *Computer Vision, Graphics and Image Processing*, 39:267–278, 1987. doi: 10.1016/S0734-189X(87)80181-0.
- [29] A. Masood and M. Sarfraz. Capturing outlines of 2d objects with bézier cubic approximation. *Image and Vision Computing*, 27:704–712, 2009. doi: 10.1016/j.imavis.2008.07.012.
- [30] B. Deffontaines C. Ichouku and J. Chorowicz. Segmentation of digital plane curves: A dynamic focusing approach. *Pattern Recognition Letters*, 17:741–750, 1996. doi: 10.1016/0167-8655(96)00015-3.
- [31] Wu-Chih Hu. Multiprimitive segmentation based on meaningful breakpoints for fitting digital planar curves with line segments and conic arcs. *Image and Vision Computing*, 23:783–789, 2005. doi: 10.1016/0167-8655(96)00015-3.
- [32] Hsin-Teng Sheu and Wu-Chih Hu. A rotationally invariant two-phase scheme for corner detection. *Pattern Recognition*, 29(5):819–828, 1996. doi: 10.1016/0031-3203(95)00121-2.
- [33] Yu-Hua Gu and T. Tjahjadi. Coarse-to-fine planar object identification using invariant curve features and b-spline modeling. *Pattern Recognition*, 33:1411–1422, 2000. doi: 10.1016/S0031-3203(99)00131-4.
- [34] R. C. González and R. E. Woods. *Tratamiento Digital de Imágenes*. Addison-Wesley Iberoamericana, S. A. y Ediciones Díaz de Santos, S. A., 1996. ISBN: 9780201625769.
- [35] J. González Jiménez. *Visión por Computador*. Paraninfo, Madrid, España, 2000. ISBN: 8428326304.
- [36] I. Pitas. *Digital Image Processing Algorithms*. International Series in Acoustics, Speech and Signal Processing. Prentice Hall, 1993. ISBN: 0131458140.
- [37] M. Sonka. *Image Processing, Analysis and Machine Vision*. Chapman y Hal. Primera Edición, 1993. ISBN: 049508252X.
- [38] D. Zhang. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004. doi: 10.1016/j.patcog.2003.07.008.
- [39] J. O’Rourke A. Melkman. On polygonal chain approximation. *G.T. Toussaint (Ed.), Computational Morphology, North-Holland, Amsterdam*, pages 87–95, 1998. doi: 10.1016/B978-0-444-70467-2.50012-6.
- [40] P. Fränti A. Kolesnikov. Polygonal approximation of closed discrete curves. *Pattern Recognition*, 40:1282–1293, 2007. doi: 10.1016/j.patcog.2006.09.002.
- [41] A. Carmona Poyato, N. L. Fernández García, R. Medina Carnicer, and F.J. Madrid Cuevas. Dominant point detection: A new proposal. *Image and Vision Computing*, 23(13):1226–1236, 2005.

- [42] M. T. Parvez and S. A. Mahmoud. Polygonal approximation of digital planar curves through adaptive optimizations. *Pattern Recognition Letters*, 31(13):1997–2005, October 2010. <http://dx.doi.org/10.1016/j.patrec.2010.06.007>.
- [43] K. S. Ray B. K. Ray. A non-parametric sequential method for polygonal approximation of digital curves. *Pattern Recognition Letters.*, 15:161–167, 1994. doi: 10.1016/0167-8655(94)90045-0.
- [44] M. Iri H. Imai. An optimal algorithm for approximating a piecewise linear function. *J. Inform. Process.*, 9:159–162, 1986.
- [45] J. C. Perez and E. Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15:743–750, 1994. doi: 10.1016/0167-8655(94)90002-7.
- [46] M. Salotti. An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters*, 22:215–221, 2001. doi: 10.1016/S0167-8655(00)00088-X.
- [47] M. Salotti. Optimal polygonal approximation of digitized curves using the sum of square deviations criterion. *Pattern Recognition*, 35:435–443, 2002. doi: 10.1016/S0031-3203(01)00051-6.
- [48] I. Dinstein A. Pikaz. Optimal polygonal approximation of digital curves. *Pattern Recognition*, 28:373–379, 1995.
- [49] K. S. Ray B. K. Ray. Determination of optimal polygon from digital curves using l1 norm. *Pattern Recognition*, 26:50–509, 1993. doi: 10.1016/0031-3203(93)90106-7.
- [50] J.T. Li J.-H. Horng. An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves. *Pattern Recognition Letters*, 23:171–182, 2002. doi: 10.1016/S0167-8655(01)00098-8.
- [51] E. Ataer-Cansizoglu et al. Contour-based shape representation using principal curves. *Pattern Recognition*, 46:1140–1150, 2013. doi: 10.1016/S0167-8655(01)00098-8.
- [52] D. H. Douglas and T. 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*, 10:112–122, 1973. doi: 10.3138/FM57-6770-U75U-7727.
- [53] A. Carmona-Poyato, N. L. Fernández-García, and R. Muñoz-Salinas. A new algorithm for dominant point detection by quasi-collinear break points supression. *Lecture Notes in Computer Science*, 23:474–484, 2010. doi: 10.1007/978-3-540-88458-3_43.
- [54] P. I. Hosur, P.I., and M. Kai-Kuang. A novel scheme for progressive polygon approximation of shape contours. *Proceedings of IEEE 3rd Workshop on Multimedia Signal Processing*, pages 309–314, 1999. doi: 10.1109/MMSP.1999.793854.
- [55] J. M. I nesta, M. Buendia, and M.A. Sart. Reliable polygonal approximations of imaged real objects through dominant point detection. *Pattern Recognition*, 31:685–697, 1998. doi: 10.1016/S0031-3203(97)00081-2.

- [56] D. G. Lowe. Three dimensional object recognition from single two dimensional images. *Artificial Intelligence*, 31:355–395, 1987. doi: 10.16/0004-3702(87)90070-1.
- [57] A. Masood. Optimized polygonal approximation by dominant point deletion. *Pattern Recognition*, 41, 2008. doi: 10.1016/j.patcog.2007.05.021.
- [58] A. Pikaz and I. Dinstein. An algorithm for polygonal approximation based on iterative point elimination. *Pattern Recognition Letters*, 16(6):557–563, 1995. doi: 10.1016/0167-8655(95)80001-A.
- [59] V. Gonzalez J. Sklansky. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12:327–331, 1980. doi: 10.1016/0031-3203(80)90031-X.
- [60] C.C. Tseng, C.J.Juan, and J.FLin H.C.Chang. An optimal line segment extraction algorithm for on line chinese character recognition using dynamic programming. *Pattern Recognition Letters*, 19:953–961, 1998. doi: 10.1016/S0167-8655(98)00071-3.
- [61] T.P. Nguyen and I. Debled-Rennesson. A discrete geometry approach for dominant point detection. *Pattern Recognition*, 44(1):32–44, 2011. doi: 10.1016/j.patcog.2010.06.022.
- [62] F. Mokhtarian and A. K. Mackworth. A theory of multiscale-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992. doi: 10.1109/34.149591.
- [63] F. Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:539–544, 1995. doi: 10.1109/34.391387.
- [64] S.C. Pei and C.N. Lin. The detection of dominant points on digital curves by scale-space filtering. *Pattern Recognition*, 25:1307–1314, 1992. doi: 10.1016/0031-3203(92)90143-7.
- [65] W. Y. Wu. Dominant point detection using bending value. *Image and Vision Computing*, 21:517–525, 2003. doi: 10.1016/S0262-8856(03)00031-3.
- [66] N. Ansari and K.W. Huang. Non-parametric dominant points detection. *Pattern Recognition*, 24:849–862, 1991. doi: 10.1016/0031-3203(91)90004-O.
- [67] G. Ramella C. Arcelli. Finding contour-based abstractions of planar pattern. *Pattern Recognition*, 26(10):1563–1577, 1993. doi: 10.1016/0031-3203(93)90161-O.
- [68] A. Bandera, C. Urdiales, F. Arrebola, and F. Sandoval. Corner detection by means of adaptively estimated curvature function. *Electron. Let*, 36(2):124–126, 2000. doi: 10.1049/el:20000177.
- [69] C. P. Chau and W. S. Siu. New nonparametric dominant point detection algorithm. *IEEE Proc. Visual Image Signal Process.*, 148(5):363–374, 2001. doi: 10.1049/ip-vis:20010576.

- [70] T. M. Cronin. A boundary concavity code to support dominant point detection. *Pattern Recognition Letters*, 20:617–634, 1999. doi: 10.1016/S0167-8655(99)00025-2.
- [71] A. Held, K. Abe, and C. Arcelli. Towards a hierarchical contour description via dominant point detection. *IEEE Trans. Syst. Man Cybern.*, 24:942–949, 1994. doi: 10.1109/21.293514.
- [72] M. S. Kankanhalli. An adaptive dominant point detection algorithm for digital curves. *Pattern Recognition Letters*, 14:385–390, 1993. doi: 10.1016/0167-8655(93)90116-U.
- [73] J. Koplowitz and S. Plante. Corner detection for chain coded curves. *Pattern Recognition*, 28:843–852, 1995. doi: 10.1016/0031-3203(94)00100-Z.
- [74] L. J. Latecki and R. Lakämper. Polygon evolution by vertex deletion, scale-space. *Lecture Notes in Computer Science*, 182:398–409, 1999. doi: 10.1007/3-540-48236-9_35.
- [75] M. Marji and P. Siy. A new algorithm for dominant point detection and polygonization of digital curves. *Pattern Recognition*, 36:2239–2251, 2003. doi: 10.1016/S0031-3203(03)00119-5.
- [76] D. K. Prasad, M.K.H. Leung, C. Quek, and S.-Y. Chor. A novel framework for making dominant point detection methods non-parametric. *Image and Vision Computing*, 30:843–859, 2012. doi: 10.1016/j.imavis.2012.06.010.
- [77] B. K. Ray and K. S. Ray. Detection of significant points and polygonal approximation of digitized curves. *Pattern Recognition Letters*, 12:443–452, 1992. doi: 10.1016/0167-8655(92)90051-Z.
- [78] B. K. Ray. An algorithm for detecting dominant points and polygonal approximation of digitized curves. *Pattern Recognition Letters*, 13:849–856, 1992. doi: 10.1016/0167-8655(92)90084-D.
- [79] P. V. Sankar and C. V. Sharma. A parallel procedure for the detection of dominant points on a digital curve. *Comput. Graphics Image Process.*, 7:403–412, 1978. doi: 10.1016/S0146-664X(78)80006-9.
- [80] D. Sarkar. A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition*, 14, 1993. doi: 10.1016/0167-8655(93)90004-W.
- [81] C. H. Teh and R. T. Chin. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859–872, 1989. doi: 10.1109/34.31447.
- [82] C. Urdiales, A. Bandera, and F. Sandoval. Non-parametric planar shape representation based on adaptive curvature functions. *Pattern Recognition*, 35:43–53, 2002. doi: 10.1016/S0031-3203(01)00041-3.

- [83] W. Y. Wu. An adaptive method for detecting dominant points. *Pattern Recognition*, 36:2231–2237, 2003. doi: 10.1016/S0031-3203(03)00087-6.
- [84] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1:244–256, 1972. doi: 10.1016/S0146-664X(72)80017-0.
- [85] A. Carmona-Poyato, R. Medina-Carnicer, F. J. Madrid-Cuevas, R. Muñoz-Salinas, and N. L. Fernández García. A new measurement for assessing polygonal approximation of curves. *Pattern Recognition*, 4(1):44 – 54, January 2011. doi: 10.1016/j.patcog.2010.07.029.
- [86] T. Kauranne A. Kolesnikov. Unsupervised segmentation and approximation of digital curves with rate-distortion curve modeling. *Pattern Recognition*, 47:623–633, 2014. doi: 10.1016/j.patcog.2013.09.002.
- [87] M. T. Parvez. Optimized polygonal approximations through vertex relocations in contour neighborhoods. *Image and Vision Computing*, 34:1–10, 2015. doi: 10.1016/j.imavis.2014.10.012.
- [88] P. L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19:9–666, 1997. doi: 10.1109/34.601253.
- [89] K. Ebisch. A correction to the douglas-peucker line generalization algorithm'. *Comp. Geosci.*, 28:995–997, 2002. doi: 10.1016/S0098-3004(02)00009-2.
- [90] J. Hershberger and J. Snoeyin. Speeding up the douglas-peucker line simplification algorithm. *Proceedings of the Fifth International Symposium on Spatial Data Handling*, 1992. doi: 10.1.1.17.6932.
- [91] P. L. Rosin. Unimodal thresholding. *Pattern Recognition*, 34:2083–2096, 2001. doi: 10.1016/S.0031-320103(00)00136-9.
- [92] S. Jeannin and M. Bober. Description of core experiments for mpeg-7 motion/shape, mpeg-7. Technical report, ISO/IEC/JTC1/SC29/WG11/MPEG99/N2690, Seoul, South Korea, March 1999.
- [93] L.J. Latecki. Mpeg-7 core experiment ce-shape-1 test set (part b). benchmarking image database for shape recognition techniques. <http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz>.
- [94] L.J. Latecki, R. Lakämper, and Ulrich Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2000. doi: 10.1109/CVPR.2000.855850.
- [95] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, December 1945. doi: 10.2307/3001968.

- [96] N.L. Fernández-García, L. Del-Moral Martínez, A. Carmona-Poyato, F.J. Madrid-Cuevas, and R. Medina-Carnicer. A new thresholding approach for automatic generation of polygonal approximations. *Journal of Visual Communication and Image Representation*, 35:155–168, February 2016. doi:10.1016/j.jvcir.2015.12.013.
- [97] A. M. Andrew. Another efficient algorithm for convex hull in two dimensions. *Information Processing Letters*, 9:216–219, 1979. doi:10.1016/0020-0190(79)90072-3.
- [98] R. L. Graham. Finding the convex hull of a simple polygon. *Journal of Algorithms*, 4:324–331, 1983. doi:10.1007/BF00993195.

Índice de figuras

1.1. Fases del procesamiento digital de imágenes	2
1.2. Extracción del contorno de un objeto de una imagen cualquiera	3
1.3. Direcciones de los códigos de cadena: (a) conectividad 4; (b) conectividad 8.	4
1.4. Códigos de cadena de (a) 4 direcciones y (b) 8 direcciones.	4
2.1. Estrategia de división de puntos (<i>split</i>)	11
2.2. Estrategia de fusión de puntos (<i>merge</i>)	13
2.3. Distancia de un punto Q al lado de la aproximación poligonal definido por los puntos P_i y P_{i+1}	17
3.1. Inconvenientes del método RDP original	20
3.2. Esquema del método <i>split</i> propuesto	22
3.3. Primera etapa del método <i>split</i> . Selección de los puntos <i>iniciales</i> del contorno	23
3.4. Segunda etapa del método <i>split</i> . Computación de los valores de significación de los puntos que no son iniciales (1)	25
3.5. Segunda etapa del método <i>split</i> . Computación de los valores de significación de los puntos que no son iniciales (2)	25
3.6. Tercera etapa del método <i>split</i> . Computación de los valores de significación de los puntos iniciales	26

3.7. Cuarta etapa del método <i>split</i> . Generación de la curva de significación normalizada	27
3.8. Quinta etapa del método <i>split</i> . Primer y segundo método de umbralización	29
3.9. Quinta etapa del método <i>split</i> . Tercer y cuarto método de umbralización .	30
3.10. Quinta etapa del método <i>split</i> . Quinto y sexto métodos de umbralización .	30
3.11. Quinta etapa del método <i>split</i> . Quinto método de umbralización. Caso especial	31
3.12. Primer experimento. Gráficas comparativas del rendimiento del nuevo método utilizando diferentes métodos de umbralización	35
3.13. Primer experimento. Aproximaciones poligonales del contorno <i>bell-7</i>	36
3.14. Primer experimento. Aproximaciones poligonales del contorno <i>octopus-14</i> .	37
3.15. Primer experimento. Aproximaciones poligonales del contorno <i>ray-17</i> . . .	38
3.16. Segundo experimento. Gráficas comparativas del rendimiento de los métodos RDP, Carmona, Masood y el método propuesto	42
3.17. Segundo experimento. Aproximaciones poligonales del contorno <i>chicken-5</i> .	43
3.18. Segundo experimento. Aproximaciones poligonales del contorno <i>device6-9</i> .	44
3.19. Segundo experimento. Aproximaciones poligonales del contorno <i>butterfly-13</i>	45
3.20. Segundo experimento. Aproximaciones poligonales del contorno <i>bell-10</i> . .	46
3.21. Segundo experimento. Aproximaciones poligonales del contorno <i>truck-07</i> .	47
3.22. Segundo experimento. Aproximaciones poligonales del contorno <i>semicircles</i>	48
4.1. Esquema del método de la estrategia <i>split</i> . Se resaltan en amarillo las etapas que se han modificado o añadido al método original.	51
4.2. Primera etapa: selección de puntos iniciales. Alternativas 1 y 2.	53
4.3. Primera etapa: selección de puntos iniciales. Alternativas 3 y 4.	54
4.4. Primera etapa: dificultades del uso del <i>casco convexo</i>	55
4.5. Segunda etapa: cálculo de la medida <i>Emax</i>	58
4.6. Segunda etapa: cálculo de la medida <i>ISE</i>	59
4.7. Segunda etapa: cálculo de la medida <i>Longitud</i>	59
4.8. Séptima etapa. Borrado de puntos superfluos.	62

4.9. Esquema del método de la estrategia <i>merge</i> . Se resaltan en amarillo las etapas que han sufrido cambios respecto del método original.	64
4.10. Segunda etapa. Cálculo de la significación de los puntos del contorno. Problemas de simetría del método <i>merge</i>	66
4.11. Tercera etapa. Cálculo de la significación acumulada	70
4.12. Primer experimento. Gráficas comparativas del rendimiento de las versiones <i>split</i>	77
4.13. Primer experimento: comparación de aproximaciones poligonales del contorno <i>bird-17</i> (N = 779 puntos) obtenidas por versiones <i>split</i>	78
4.14. Primer experimento: comparación de aproximaciones poligonales del contorno <i>cattle-6</i> (N = 1744 puntos) obtenidas por versiones <i>split</i>	79
4.15. Segundo experimento. Gráficas comparativas del rendimiento de las versiones <i>merge</i>	83
4.16. Segundo experimento: comparación de aproximaciones poligonales del contorno <i>butterfly-20</i> (N = 1865 puntos) obtenidas por versiones <i>merge</i>	84
4.17. Segundo experimento: comparación de aproximaciones poligonales del contorno <i>fish-7</i> (N = 866 puntos) obtenidas por versiones <i>merge</i>	85
4.18. Comparación de aproximaciones poligonales del contorno del contorno <i>crown-12</i> (N = 335 puntos) obtenidas por versiones <i>split</i> y <i>merge</i>	87
4.19. Comparación de aproximaciones poligonales del contorno del contorno <i>device3-7</i> (N = 1547 puntos) obtenidas por versiones <i>split</i> y <i>merge</i>	88
5.1. Esquema del método optimizado de la estrategia <i>split</i>	91
5.2. Esquema del método optimizado de la estrategia <i>merge</i>	92
5.3. Proceso de optimización de un punto dominante.	93
5.4. Aplicación del proceso de optimización en las versiones <i>split</i>	94
5.5. Primer experimento. Gráficas comparativas del rendimiento de las versiones <i>split</i> optimizadas	100
5.6. Primer experimento: comparación de aproximaciones poligonales del contorno <i>car-04</i> (N = 424 puntos) obtenidas por las versiones <i>split</i> optimizadas	101
5.7. Primer experimento: comparación de aproximaciones poligonales del contorno <i>cellular_phone-17</i> (N = 857 puntos) obtenidas por las versiones <i>split</i> optimizadas	102

- 5.8. Primer experimento: comparativa del ajuste de las versiones *split* 4-2-1-6 y 4-1-1-5 optimizadas respecto a las versiones *split* 4-2-1-6 y 4-1-1-5 originales 103
- 5.9. Primer experimento: comparativa del ajuste de las versiones *split* 2-1-1-6 y 2-1-2-5 optimizadas respecto a las versiones *split* 2-1-1-6 y 2-1-2-5 originales 104
- 5.10. Segundo experimento. Gráficas comparativas del rendimiento de las versiones *merge* optimizadas 109
- 5.11. Segundo experimento: comparación de aproximaciones poligonales del contorno *bell-14_0* (N = 712 puntos) obtenidas por versiones *split* 110
- 5.12. Segundo experimento: comparación de aproximaciones poligonales del contorno *butterfly-16_0* (N = 1819 puntos) obtenidas por versiones *split* . . . 111
- 5.13. Segundo experimento: comparativa del ajuste de las versiones *merge* 1-5 y 2-5 optimizadas respecto a las versiones *merge* 1-5 y 2-5 originales 112
- 5.14. Segundo experimento: comparativa del ajuste de las versiones *merge* 1-6 y 2-6 optimizadas respecto a las versiones *merge* 1-6 y 2-6 originales 113
- 5.15. Comparación de aproximaciones poligonales del contorno *apple-19* (N = 733 puntos) obtenidas por versiones optimizadas de las estrategias *split* y *merge* 115
- 5.16. Comparación de aproximaciones poligonales del contorno del contorno *carriage-02* (N = 742 puntos) obtenidas por versiones optimizadas de las estrategias *split* y *merge* 116

Índice de tablas

3.1. Primer experimento. Comparativa del rendimiento del nuevo método utilizando diferentes métodos de umbralización	34
3.2. Test de Wilcoxon de los métodos de umbralización propuestos	39
3.3. Segundo experimento. Comparativa del rendimiento del nuevo método utilizando diferentes métodos de umbralización	41
4.1. Tabla de alternativas <i>split</i>	72
4.2. Primer experimento. Comparativa del rendimiento de las versiones de la estrategia <i>split</i>	76
4.3. Tabla de alternativas <i>merge</i>	79
4.4. Segundo experimento. Comparativa del rendimiento de las alternativas de la estrategia <i>merge</i>	82
4.5. Comparativa del rendimiento de las mejores versiones <i>split</i> y <i>merge</i>	86
5.1. Primer experimento. Comparativa del rendimiento de las versiones de la optimización <i>split</i>	99
5.2. Segundo experimento. Comparativa del rendimiento de las versiones de la optimización <i>merge</i>	108
5.3. Comparativa del rendimiento de las mejores versiones <i>split</i> y <i>merge</i>	114

