

Capítulo 2

Correspondencia de formas 2D y cálculo de la forma media

Este capítulo presenta la manera de solucionar el problema de comparar formas 2D, que tenemos dentro del análisis de texturas estructuradas. Dentro de este análisis necesitamos encontrar formas 2D similares que se emplazan siguiendo una regla de emplazamiento similar. Para poder determinar si dos formas son similares necesitamos una representación de las mismas y una función de similitud sobre ellas. Explicamos diversos métodos que existen en la literatura para representar formas 2D y compararlas y nos centramos en los utilizados en este trabajo. En concreto representamos las formas como cadenas cíclicas aumentadas con atributos que representan sus características geométricas, y las comparamos utilizando la distancia de edición de cadenas. Otro objetivo del trabajo es inferir la estructura de una textura estructurada, para ello necesitamos determinar la forma que representa al textón que está formando la textura, por ello necesitamos calcular la forma media de un conjunto dado. Como en nuestro caso las formas están representadas por cadenas, el problema queda definido como el cálculo de la cadena media. Por ello presentamos un método para el cálculo de la cadena media en el marco concreto en que esas cadenas representan formas.

2.1 Las formas 2D en este trabajo

Hemos visto en el capítulo 1 que segmentar texturas estructuradas dentro del marco de los planos de arquitectura significa encontrar conjuntos de elementos similares, en nuestro caso polígonos, que estén emplazados siguiendo una misma regla de emplazamiento. Para poder saber si dos polígonos son similares o no necesitamos tener una estructura que los represente y definida una distancia entre ellos. Además dado un conjunto de polígonos similares que formen una textura, es necesario poder calcular un representante para utilizarlo en la modelización de la textura. Por todo ello en este capítulo vamos a presentar la estructura de representación de las formas 2D que

son polígonos, su función de distancia y el método para calcular la forma media de un conjunto dado.

2.2 Métodos existentes para resolver el problema de correspondencia de formas.

Existen diversas técnicas para resolver el problema en el análisis de formas 2D. Pero se pueden clasificar en dos enfoques básicos: Los métodos estadísticos y los estructurales, para más información ver [24, 97, 99].

En el *método estadístico* se extrae un conjunto de medidas de las propiedades de los patrones, llamadas características, y el reconocimiento de patrones puede verse como una partición del espacio de características. En el *método estructural* la estructura de los patrones se representa como una composición de unidades de estructura, llamadas primitivas, y los patrones se comparan poniendo en correspondencia sus representaciones estructurales, o analizando sintácticamente la representación según un conjunto de reglas sintácticas. En estos métodos estructurales la selección de las primitivas es un punto clave, y a veces el ratio de reconocimiento depende del resultado de la selección de las mismas. Normalmente, una selección de primitivas simples hace que el proceso de segmentación sea más sencillo, pero que el proceso de reconocimiento incremente su complejidad y viceversa. Una manera de representar la estructura de un patrón es por medio de una cadena, este método permite representar una estructura de 2D en una sola dimensión, y la forma de comparar dos patrones representados por cadenas es la puesta en correspondencia de cadenas.

2.3 Correspondencia de cadenas

En esta sección vamos a explicar la correspondencia de cadenas. Primero se necesitan definir algunos conceptos, después en el apartado 2.3.2 se explicará cómo representar formas 2D por medio de cadenas, para finalizar en el apartado 2.3.3 explicando algunos métodos de comparación de cadenas.

2.3.1 Algunas definiciones de la teoría de cadenas

La teoría de cadenas tiene algunas definiciones básicas que vamos a explicar en este apartado y las usaremos a lo largo de toda la disertación.

Definición 2.1 Sea Σ un alfabeto de símbolos, Σ^* el conjunto de todas las cadenas finitas sobre Σ y $A = a_1 \dots a_n \in \Sigma^*$, $B = b_1 \dots b_m \in \Sigma^*$ dos cadenas; $n, m \geq 0$. La distancia entre A y B se define en términos de operaciones básicas de edición, que son necesarias para transformar A en B con el mínimo coste. Convencionalmente se definen tres operaciones de edición:

1. **substitución** de un símbolo $a \in \Sigma$ en A por un símbolo $b \in \Sigma$ en B , denotado como $a \rightarrow b$;
2. **inserción** de un símbolo $b \in \Sigma$ en B , denotado como $\lambda \rightarrow b$;

3. **borrado** de un símbolo $a \in \Sigma$ en A , denotado como $a \rightarrow \lambda$;

donde λ denota la cadena vacía.

Definición 2.2 Una *secuencia de edición* S se define como una secuencia ordenada de operaciones de edición s_1, \dots, s_p .

Definición 2.3 Sea c una función de coste que asigna un número real no negativo $c(s)$ a cada operación de edición s . El *coste* de una secuencia de edición S se define como:

$$c(S) = \sum_{i=1}^p c(s_i) \quad (2.1)$$

y la *distancia de edición* entre las cadenas A y B se define como

$$d(A, B) = \min\{c(S) : S \text{ es una secuencia de operaciones de edición que transforman } A \text{ en } B\}. \quad (2.2)$$

Definición 2.4 Sea $A^{(i)}$ la cadena A desplazada cíclicamente i posiciones, es decir se puede representar como $A^{(i)} = a_{i+1} \dots a_n a_1 \dots a_i$. Dos cadenas A y A' de longitud n se dirá que son *equivalentes* si $A' = A^{(i)}$ para algún $1 \leq i \leq n$. Esto define una relación de equivalencia sobre Σ^* .

Definición 2.5 La clase de equivalencia de una cadena A , será denotada como \overline{A} , y se llamará *cadena cíclica*.

Definición 2.6 La *distancia de edición cíclica* entre dos cadenas A y B se define como

$$d_c(A, B) = \min\{d(A^{(i)}, B^{(j)}) : i = 1, \dots, n \ j = 1, \dots, m\}. \quad (2.3)$$

Definición 2.7 Sea A una cadena $A = a_1 a_2 \dots a_n$, diremos que una cadena $B = b_1 \dots b_m$, $m \leq n$ es una *subcadena* de A si $A = a_1 \dots a_{i-1} B a_{j+1} \dots a_n$. La subcadena B se denotará como $A_{i,j} = a_i \dots a_j$, $1 \leq i, j \leq n$.

2.3.2 Representando una forma por medio de una cadena

El contorno cerrado de una forma 2D se puede representar por medio de una cadena, para la que es importante seleccionar un conjunto apropiado de primitivas, que se utilizarán como símbolos. Bunke y Buhler [18] propusieron un esquema basado en píxeles donde cada símbolo representa la curvatura de cada punto del contorno. También existen diversas aproximaciones que han sido presentadas para aproximar un contorno por medio de un polígono [69, 98, 99]. El uso de un conjunto discreto de símbolos hace que, en algunos casos, el polígono no pueda describir la forma de una manera precisa. Tsai y Yu [98] resolvieron este problema por medio de las *cadena de atributos*, es decir, cadenas cuyos símbolos están aumentados con valores numéricos, por ejemplo, la longitud o dirección de los segmentos de línea. En la discusión que realizamos a continuación, y sin perder generalización por ello, consideraremos que las formas están aproximadas por polígonos. Cada segmento de línea s_i se representa por su longitud l_i y su ángulo ϕ_i . El atributo ángulo se calcula con respecto a la horizontal. Esta representación está explicada con mayor detalle en el Apartado 2.4.3.

2.3.3 Diversas aproximaciones de la correspondencia de cadenas

El algoritmo de Wagner y Fischer [106] calcula $d(A, B)$ en $O(nm)$ de tiempo y espacio. Es un procedimiento de programación dinámica que calcula los elementos de una matriz D de $(n + 1) \times (m + 1)$, como se ilustra en la Fig. 2.1. Sean A y B dos cadenas de atributos. El valor $D(i, j)$ corresponde con el coste de edición $d(A_{1,i}, B_{1,j})$. Este valor $D(i, j)$ se calcula, por medio de la minimización del coste, a partir de sus tres predecesores y la operación de edición correspondiente. Es decir, yendo de $D(i - 1, j - 1)$ a $D(i, j)$ (substitución de a_i por b_j), de $D(i - 1, j)$ a $D(i, j)$ (borrado de a_i), o de $D(i, j - 1)$ a $D(i, j)$ (inserción de b_j). Para cada elemento de la matriz se guarda un puntero del predecesor seleccionado. De esta forma se puede encontrar la secuencia de edición, en orden inverso, que corresponde con $d(A, B)$. El valor de distancia se guarda en $D(n, m)$.

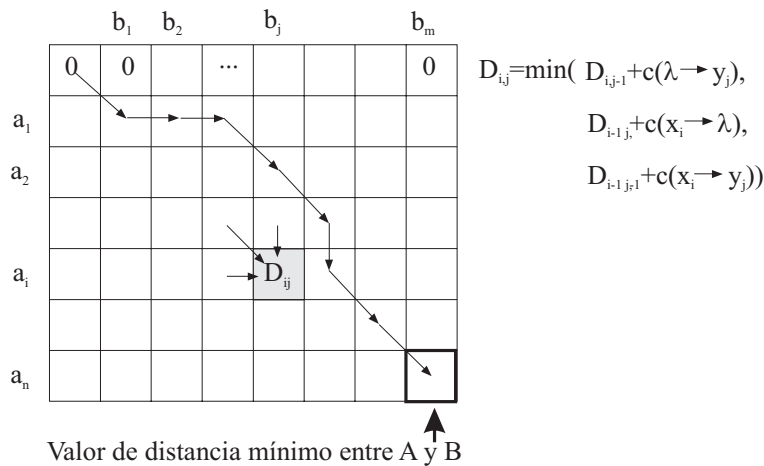


Figura 2.1: Matriz D calculada mediante la edición y puesta en correspondencia de cadenas con 3 operaciones básicas de edición.

Un problema de representación de formas 2D mediante cadenas es determinar cual es el símbolo inicial de la cadena, ya que si tenemos un contorno cerrado, no está definido de forma única. Para resolver este problema de manera exhaustiva se puede calcular la distancia de edición entre una cadena y todos los posibles desplazamientos de la otra y escoger el que hace la distancia mínima. Pero este procedimiento consumiría un tiempo del orden de $O(nm^2)$. Bunke y Buhler [18] utilizaron un procedimiento de puesta en correspondencia de cadenas cíclicas que funciona con un orden de $O(nm \log m)$. Se basa en el cálculo de la distancia básica de cadenas entre A y B^2 , donde B^2 es la cadena BB . Se demostró que $d_c(A, B)$ es igual a $d(A, B')$ cuando B' es la subcadena de B^2 que se parece más a A . Así para comparar dos cadenas cíclicas de atributos A y B se calcula la distancia de edición cíclica entre ellas $d_c(A, B)$ de forma similar a la Definición 2.6 pero desplazando sólo B . Para calcularlo se necesita una matriz D de $(n + 1) \times 2 * m + 1$, como se ilustra en la Fig. 2.2. Después de una inicialización adecuada se calcula el valor de cada $D(i, j)$ como en el algoritmo básico

pero ahora la distancia de edición cíclica se calcula cogiendo el valor mínimo de entre los que aparecen entre las posiciones de la tabla $D(n, m + 1)$ y $D(n, 2 * m)$.

$$D_{ij} = \min(D_{i,j-1} + c(\lambda \rightarrow y_j), D_{i-1,j} + c(x_i \rightarrow \lambda), D_{i-1,j-1} + c(x_i \rightarrow y_j))$$

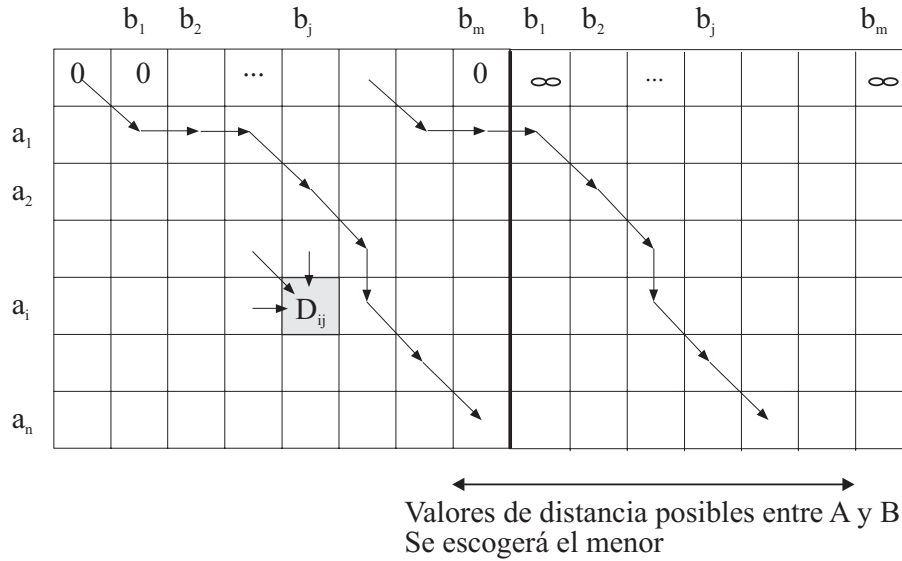


Figura 2.2: Matriz D calculada usando el algoritmo de distancia de edición de cadenas usando cadenas cíclicas.

Otro problema común en la puesta en correspondencia de cadenas para el reconocimiento de formas 2D es que en las imágenes los bordes de las formas a menudo contienen algunas distorsiones. Para solucionar este problema Tsai y Yu [98] extendieron las tres operaciones básicas con una *operación de fusión*.

Definición 2.8 Sea $A \in \Sigma^*$ definimos la *operación de fusión* denotada como $A_{i,j} \rightarrow a$ como la operación que transforma la subcadena $A_{i,j}$ en un símbolo $a \in \Sigma$. De forma equivalente a las tres operaciones básicas, la operación de fusión implicará una función de *coste de fusión* denotada como $c(A_{i,j} \rightarrow a)$.

Esta operación permite calcular *substituciones de bloques*, como hicieron Bunke y Buhler [18], es decir una substitución de toda una secuencia de símbolos por otra.

Definición 2.9 Una *substitución de bloque* se denota como $A_{i,j} \rightarrow B_{k,l}$. Para realizar esta substitución fusionamos $A_{i,j}$ en un símbolo a y la subcadena $B_{k,l}$ en un símbolo b . La substitución en bloque tiene asociado un coste que implica calcular los costes de fusión explicados en la definición anterior $c(A_{i,j} \rightarrow a)$ y $c(B_{k,l} \rightarrow b)$ y además el *coste de substitución* $c(a \rightarrow b)$.

Así el valor $D(i, j)$ se calcula, por medio de la minimización del coste, pero no sólo a partir de sus tres predecesores:

38CORRESPONDENCIA DE FORMAS 2D Y CÁLCULO DE LA FORMA MEDIA

- $D(i - 1, j - 1)$ (substitución de a_i por b_j)
- $D(i - 1, j)$ (borrado de a_i)
- $D(i, j - 1)$ (inserción de b_j)

Sino por un conjunto de celdas que vamos a dividir en tres categorías:

- La primera es el vector que va desde $D(i - l, j)$ a $D(i - 1, j)$, ir de una de estas celdas $D(k, j)$, siendo $i - l \leq k \leq i - 1$, hasta $D(i, j)$ significa *borrar* la cadena $A_{k,i-1}$.
- La segunda es el vector que va desde $D(i, j - l)$ a $D(i, j - 1)$, ir desde una de esas celdas $D(i, k)$, siendo $j - l \leq k \leq j - 1$, hasta $D(i, j)$ significa la *inserción* de la subcadena $B_{k,j-1}$.
- La tercera es la matriz que va desde $D(i - l, j - l)$ hasta $D(i - l, j - 1)$, ir desde una de esas celdas $D(o, p)$, siendo $i - l \leq o \leq i - 1$ y $j - l \leq p \leq j - 1$, hasta $D(i, j)$ significa la *substitución* de la secuencia completa de símbolos por otra ($A_{o,i-1} \rightarrow B_{p,j-1}$).

Esta tabla de distancia de edición se muestra en la Fig. 2.3, y el algoritmo se detalla en 2.3.1.

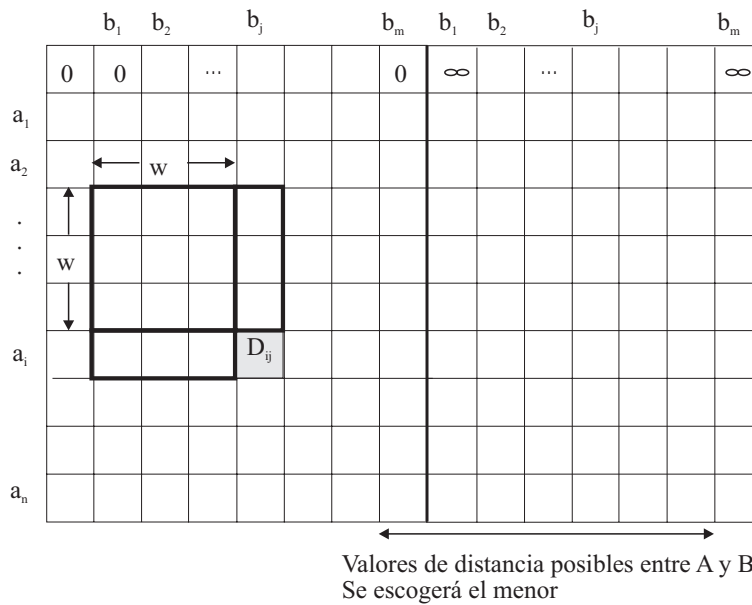


Figura 2.3: Matriz D calculada por la puesta en correspondencia mediante la distancia de cadenas con cadenas cíclicas y la operación de edición de fusión.

Para definir nuestra medida de similitud entre dos formas asumimos, sin pérdida de generalidad, que las formas están aproximadas por polígonos. Cada segmento de línea

Algoritmo 2.3.1 Puesta en correspondencia de cadenas cíclicas

entrada: Dos cadenas $X = x_1 \dots x_n$ y $Y = y_1 \dots y_n$

salida: La distancia de edición cíclica $d_c(X, Y) = d_c(X, Y')$ tal que Y' es la subcadena de Y que hace mínima la distancia de edición a X

método:

$D(0, 0).coste := 0;$

Para $i = m + 1$ hasta $2m$ **hacer**

$y_i := y_{i-m}$

FinPara

Para $i = 1$ hasta m **hacer**

$D(0, i).coste := 0$

FinPara

Para $i = m + 1$ hasta $2m$ **hacer**

$D(0, i).coste := \infty$

FinPara

Para $i = 1$ hasta n **hacer**

$D(0, i).coste := D(i - 1, 0).coste + c(x_i \rightarrow \lambda)$

FinPara

Para $i = 1$ hasta n **hacer**

Para $j = 1$ hasta $2m$ **hacer**

$D(i, j).coste := \min\{D(i - k, j - l) + c(x_{i-k+1, i} \rightarrow x') + c(y_{j-l+1, j} \rightarrow y') + c(x' \rightarrow y') : k, l = 1, \dots, m\};$

$D(i, j).anterior = (i - k', j - l');$

donde k' y l' son los valores de k y l donde estaba el mínimo en el estado previo.

FinPara

FinPara

$d_c(X, Y) := \min\{D(n, i), i = m + 1, \dots, 2m\};$

s_i se representa por su longitud l_i y su ángulo ϕ_i . El atributo del ángulo se calcula con respecto a una orientación de referencia que en nuestro caso es la horizontal. Así nuestra medida de similitud entre polígonos tiene en cuenta la orientación, pero nuestra aplicación necesita esta característica. Para hacer que la medida de similitud sea invariante a rotación, el ángulo ϕ debería ser definido como la diferencia angular entre el segmento *iésimo* y el previo a éste. Un ejemplo de una forma poligonal representada por una cadena de atributos se presenta en la Fig. 2.4. El algoritmo de distancia de cadenas que se adopta para comparar dos cadenas es el que las considera cíclicas y utiliza la operación de fusión como se muestra en la Fig. 2.3. Los costes de las operaciones de edición se definen como una suma ponderada de un coste según el ángulo y un coste en función de la longitud, de forma similar a como se hace en [99]. En el trabajo presentado se ha utilizado la siguiente función de coste, siendo $a_1 a_2$ una cadena que se fusiona en un símbolo a , para el que se definen los atributos de longitud l_a y ángulo ϕ_a de la siguiente manera:

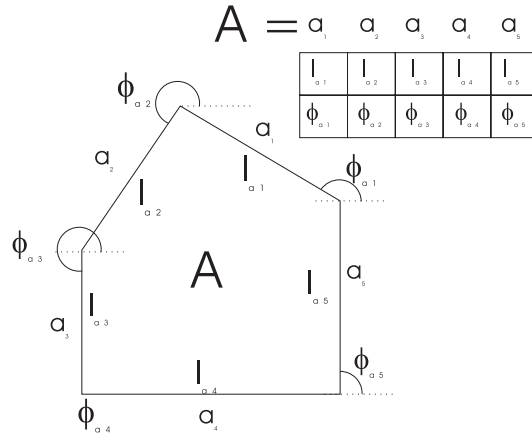


Figura 2.4: Forma representada por medio de una cadena de atributos. Los atributos son la longitud l y el ángulo ϕ .

$$l_a = l_{a_1 + a_2} \tag{2.4}$$

$$\phi_a = \phi_{a_1 + a_2} \tag{2.5}$$

donde \vec{a}_1 es el vector definido por el segmento a_1 y \vec{a}_2 es el vector definido por el segmento a_2 .

Una vez se ha definido el símbolo a , se calcula la función de coste usando la siguiente ecuación:

$$c(a_1 a_2 \rightarrow a) = \frac{|\phi_a - \phi_1| l_1}{2\pi l_a} + \frac{|\phi_a - \phi_2| l_2}{2\pi l_a} \tag{2.6}$$

Las ecuaciones previas se pueden extender de forma iterativa para fusionar sub-cadenas con más de dos símbolos. Se debe destacar que la substitución, borrado e

inserción son casos particulares de la substitución de bloques usando la operación de fusión.

Así el coste total de substituir una secuencia completa de símbolos por otra ($A_{i,j} \rightarrow B_{k,l}$) se calcula de la siguiente forma:

$$c(A_{i,j} \rightarrow B_{k,l}) = c(A_{i,j}, a') + c(B_{k,l}, b') + c(a', b') \quad (2.7)$$

2.4 Cálculo de la forma media

2.4.1 Introducción

En el campo del reconocimiento de formas aparece a menudo la necesidad de calcular una forma que represente un conjunto de formas similares, es decir una forma media. La forma media se puede calcular como la forma más cercana a todas las demás, usando una cierta distancia, o como la forma que tiene como valores en sus características el valor medio de las características de las otras. Una forma se puede representar por medio de un grafo, un árbol, una cadena, etc. La forma media de un conjunto dado es necesaria para representar al conjunto tanto en el aprendizaje de prototipos como en la inferencia de la estructura en el análisis de las agrupaciones de conjuntos de estructuras regulares. Existen dos enfoques básicos en el cálculo de la forma media, el primero es el estadístico, donde una forma está representada por un vector de características y calcular la forma media representa calcular la media numérica de las mismas, mientras que el segundo es el estructural, donde el cálculo de la forma media significa calcular una estructura 2D media, este segundo enfoque es el abordado en este trabajo y tiene una mayor complejidad.

Desde el punto de vista estructural, una forma puede representarse usando diferentes estructuras, por ello en los párrafos siguientes vamos a introducir algunos trabajos que calculan la forma media utilizando diferentes representaciones y técnicas para resolver el problema. En los dos primeros ejemplos utilizan un conjunto de características y de técnicas de modelos deformables, el tercer ejemplo usa una estructura multiescala convexa-cóncava y programación dinámica para la estrategia de puesta en correspondencia, luego se presentan tres ejemplos que usan una representación de grafo y al final se describen algunos ejemplos que utilizan cadenas. Debemos hacer notar que cuando se utiliza una cadena para representar una forma el problema de calcular la forma media es equivalente al problema del cálculo de la cadena media. Como nuestro trabajo se basa en esta representación debemos también presentar algunos trabajos que calculen la cadena media, aunque esta cadena represente solo una tira de símbolos, y no una forma, en los cuales nos hemos inspirado para realizar este algoritmo.

Valveny en [102] usa modelos deformables. Su objetivo es el reconocimiento de símbolos gráficos, y con este fin infiere un modelo para cada símbolo gráfico a partir de un conjunto de símbolos deformados, para reconocerlos en documentos gráficos. Las formas se representan por medio de un conjunto de líneas $S = \{l_1, l_2, \dots, l_n\}$. El proceso de aprendizaje es supervisado. Al principio tiene predefinido un modelo, y

ajusta las imágenes a este modelo predefinido. La representación aprendida del modelo se calcula como una media de las representaciones de todas las imágenes. Para ello se calcula la varianza de la probabilidad de deformación de cada línea a partir de la diferencia entre la representación de cada forma y el nuevo modelo del símbolo.

Cootes, Taylor y Graham en [27] describen un método para construir modelos aprendiendo patrones de variabilidad a partir de un conjunto de imágenes que están correctamente anotadas. Estos patrones están representados por Modelos de Distribución de Puntos y se usan para la búsqueda de imágenes en un algoritmo de refinamiento interactivo. Representan una imagen por medio de un conjunto de puntos etiquetados, los cuales se emplazan de forma manual en el conjunto de ejemplos, luego alinean el conjunto de entrenamiento para ser capaces de comparar los puntos equivalentes. Luego a partir de los puntos equivalentes calculan sus posiciones medias y un conjunto pequeño de nodos de variación que describen cómo puede cambiar el objeto.

Ueda y Suzuki en [101] adquieren un modelo visual generalizando la estructura convexa/cóncava multiescala de una clase de formas para usarlo después para poner en correspondencia formas. Una forma se representa a diferentes escalas, en cada una hay una lista de puntos donde la forma cambia de convexa a cóncava o viceversa. La escala más fina tiene más puntos y la más basta menos, y todos los puntos de la forma basta están en la fina. Para obtener el modelo a partir de un conjunto de N formas de la misma clase, primero calculan las formas a diferentes escalas. Luego obtienen los resultados de correspondencia poniendo en correspondencia uno con todos los demás, otro con todos los demás, y así N veces. Los resultados de la puesta en correspondencia de uno y todos los demás se integran en todos los casos, teniendo al final N resultados de integración con la misma estructura convexo-cóncava, que es el modelo de la clase.

Bunke, Jiang y Münger en [20, 46] presentan y comparan dos maneras de calcular el grafo mediana generalizado, una búsqueda combinatoria y dos versiones de un algoritmo genético. Definen el grafo mediana generalizado de un conjunto de grafos como el grafo, de entre ellos, con una distancia de edición menor a todos los grafos del conjunto. Para medir la distancia de edición definen seis operaciones de edición, inserción, borrado y sustitución de nodos y las mismas para las aristas. En la búsqueda combinatoria parten del hecho que el grafo mediana generalizado está limitado de forma teórica por el número total de nodos de todos los grafos de entrada juntos, así que pueden encontrar la solución examinando todas las puestas en correspondencia múltiples que impliquen grafos no mayores que ese número de nodos. Usan un algoritmo basado en el A^* , donde las posibles puestas en correspondencia múltiples se usan como los estados del espacio de búsqueda, y representan candidatos para ser el grafo mediana generalizado. El coste de la puesta en correspondencia múltiple es el coste que se ha de minimizar. Al principio la búsqueda empieza comprobando todas las puestas en correspondencia múltiples posibles que impliquen un solo nodo, luego se selecciona el menos costoso, si se completa la puesta en correspondencia entonces ésa es la solución, sino se añade un nodo más. En la solución del algoritmo genético

definen un cromosoma como el vector de enteros que tiene su longitud igual al número total de nodos de los diferentes grafos de entrada. Cada nodo tiene su posición correspondiente en el vector y el entero n que aparece en ella indica, si $n = 0$ que el nodo se ha insertado en el grafo candidato, si $n \geq 0$ el nodo ha sido substituido por el nodo candidato n . Se define un operador de crossover de un solo punto que corta, de los dos cromosomas que van a ser cruzados, una cola de una cierta longitud, la misma para ambos, que es seleccionada de forma aleatoria. Luego se intercambian esas colas, pero como el cromosoma puede ser inconsistente han de comprobarlo y cambiarlo en tal caso. El operador de mutación cambia aleatoriamente los valores en el vector, en este caso el cromosoma vuelve a poder ser inconsistente, así que se vuelve a comprobar que no lo sea y a cambiarlo en caso necesario. El algoritmo genético se puede inicializar de dos formas. Una es completamente aleatoria (G.A. Random), la otra se inicializa modificando un conjunto de grafos de entrada (G.G. Ini). Concluyen que el algoritmo genético da unas soluciones buenas y que las encuentra rápidamente. En el caso del G.A. Ini. las soluciones que encuentran en todas las pruebas son las correctas, así que además de ser mucho más rápido que el algoritmo A^* generalmente encuentra soluciones exactas.

Bunke y Kandel en [19] presentan una disertación teórica sobre la computación y las características de los grafos medios calculados a partir de dos grafos. Como para calcular el grafo medio necesitan comparar grafos, utilizan operaciones para transformar un grafo en el otro. Estas operaciones son: la inserción, borrado de un nodo o substitución de un nodo por otro. Cada operación tiene asociado un coste, y la distancia entre grafos se calcula en función del coste de transformar un grafo en otro usando estas operaciones de edición. En este entorno, la media de dos grafos es otro grafo, que minimiza la suma de costes al transformarlo en los dos grafos de los que se está calculando la media.

En la literatura existen métodos que calculan la cadena media de un conjunto de cadenas dado [70, 52, 54, 66]. Sabemos que una forma 2D se puede representar por medio de una cadena cíclica. Aunque el cálculo de la cadena media no es directamente aplicable al cálculo de la forma media, nos proporciona ideas útiles para determinar cómo podemos calcular la forma media, cuando ésta se representa por medio de una cadena.

Kruskal en [54] presenta un algoritmo para calcular la cadena media de tres cadenas que representan moléculas. En ese trabajo necesitaban establecer la homología entre tres moléculas a , b , c . Para establecer esta homología necesitan primeramente reconstruir la secuencia interna d y luego la homología de d con cada una de las a , b , c , y todo ello proporcionará de una forma indirecta la homología entre a , b , y c . Para construir d y calcular la homología entre d y cada una de las a , b , y c usan un algoritmo básico para calcular la distancia de edición entre cadenas, pero añaden en cada paso de la comparación el cálculo del carácter que hace mínima la distancia entre cada uno de los caracteres que han sido alineados de a , b y c . De esta forma construyen d y la homología entre d y a , b , c .

Lopresti y Zhou en [66] trabajan en el reconocimiento óptico de caracteres (OCR),

donde una cadena media se puede calcular a partir de un conjunto de cadenas posiblemente incorrectas, cada una con un error diferente, para poder encontrar la cadena correcta que está impresa en una página. En este trabajo necesitan la cadena media para corregir los errores en los resultados. El OCR es un proceso para obtener una cadena reconocida R a partir de una imagen I que representa una cadena de texto fuente S . A veces S y R no son exactamente lo mismo, para resolver este problema proponen procesar la misma página con distintos clasificadores de OCR, obteniendo de esta forma varios R_i . Cada R_i presentará un conjunto de errores diferente. A partir de las diferentes cadenas candidatas R_i , determinan una secuencia de consenso usando la distancia de edición entre dos de ellas, que ha sido explicada en el Apartado 2.3.3. Sea Σ el alfabeto de símbolos para construir las cadenas S , y R_i , y $d(C, R_i)$ la distancia de edición entre dos cadenas. El valor de la suma de distancias de una secuencia de cadenas $C \in \Sigma$, que hace mínima esta suma, a cada una de las secuencias de cadenas del conjunto (R_1, \dots, R_N) , se llama $D(R_1, \dots, R_N)$. El conjunto de todas las secuencias de consenso se llama $Cons(R_1, \dots, R_N)$ y es el conjunto de secuencias de cadenas $C_i \in \Sigma$ cuyas sumas de distancias de ellas a todas las cadenas en (R_1, \dots, R_N) es equivalente a $D(R_1, \dots, R_N)$. Es decir, es el conjunto de secuencias de cadenas que minimizan la distancia de edición entre ellas y todas las R_i . O para decirlo en otras palabras, la cadena media de todas las R_i . Esta cadena media se basa en el algoritmo presentado por Kruskal en [54].

Casacuberta y de Antonio en [22] calculan la cadena media usando un algoritmo *greedy* que calcula la cadena mediana símbolo a símbolo. Se basa en el algoritmo del cálculo de la distancia de Levenshtein. Intentan colocar todos los símbolos de su alfabeto en cada posición de la cadena media y seleccionan el mejor. Debemos notar que este algoritmo espera un alfabeto finito. Usan la cadena mediana para encontrar una topología adecuada para modelos acústicos, pero sugieren que puede ser útil para el análisis de agrupación (o clustering) de patrones sintácticos. En el artículo usan cadenas que son tiras de caracteres pero no explican como adaptarían su algoritmo en caso de que las cadenas representaran patrones 2D.

Existen básicamente dos problemas a la hora de adaptar el cálculo de la cadena media general, al cálculo de la cadena media cuando ésta representa una forma. Primero debemos seleccionar un símbolo inicial de la cadena, que no está definido de forma única cuando el contorno está cerrado, lo cual quiere decir que la cadena es cíclica, al contrario que en las cadenas habituales de caracteres. Habitualmente se utilizan alfabetos finitos de símbolos integrantes de las cadenas. Sin embargo, en determinados problemas, como en nuestro caso, para una mayor precisión de la codificación es necesario utilizar cadenas de atributos y, en consecuencia, alfabetos infinitos. Esto implica que el cálculo de la cadena media debe ser reformulado a partir de los anteriores enfoques.

En el cálculo de la forma media que presentamos aquí, las formas están representadas por cadenas cíclicas, así que la forma media es en realidad la cadena media. Modificamos el algoritmo de la distancia de edición de cadenas para calcular también la cadena media. Con la distancia de cadenas tenemos una correspondencia entre

segmentos de recta o conjuntos de segmentos de recta entre las dos cadenas que estamos comparando. A partir de esta correspondencia calculamos los segmentos de recta medios que están formando la forma media, tratándolos como si fueran funciones por trozos y calculando la forma media como una media entre funciones. La transformación a partir de un conjunto de segmentos de recta a una función por trozos se resuelve de dos formas: La primera se presenta en la Subsección 2.4.3 y transforma los segmentos en funciones usando sus valores x como parámetro común, la llamaremos Método-X. La segunda se explica en la Subsección 2.4.3 y transforma los segmentos en una función usando la longitud relativa de los segmentos como parámetro común, la llamaremos Método-L.

En las siguientes subsecciones vamos a introducir el concepto básico de la cadena media y nuestra aproximación para solucionar el problema cuando dos cadenas que representen formas 2D. Después proponemos un algoritmo para calcular la cadena media cuando ésta represente formas 2D aproximadas por polígonos. Para finalizar realizamos algunas observaciones sobre el algoritmo.

2.4.2 Definición formal de la cadena media

Uno de los primeros intentos de resolver el problema de la cadena media fue el propuesto por Kruskal [54]. Más tarde, Lopresti y Zhou [66] usaron un algoritmo similar para calcular la media de varias cadenas que habían sido obtenidas por un OCR. Nuestro algoritmo se inspira en estos dos trabajos previos, pero nosotros trabajamos con cadenas que representan formas 2D. Por ello debe tenerse en cuenta la información geométrica que expresan.

Para empezar debemos distinguir dos conceptos: la *cadena mediana de conjunto*, en inglés *set median string*, y la *cadena media*, en inglés *media string*, también llamada *cadena mediana generalizada*, en inglés *generalized median string*.

Definición 2.10 La *cadena mediana de conjunto* SM de un conjunto de cadenas $S = \{S^1, \dots, S^N\}$ se define, en términos de la distancia de edición, como la cadena del conjunto S que minimiza el coste combinado de editar SM en cada una de las cadenas S^i . El *coste de edición mínimo* de transformar SM en cada una de las S^i , se denota como $SD_c(S^1, \dots, S^N)$ y se define como sigue:

$$SD_c(S^1, \dots, S^N) \equiv \min_{SM \in S} \sum_{i=1}^N d_c(SM, S^i) \quad (2.8)$$

donde $d_c(SM, S^i)$ representa la distancia de edición cíclica entre SM y S^i .

Definición 2.11 La *cadena media* M o la *cadena mediana generalizada* de un conjunto de cadenas $S = \{S^1, S^2, \dots, S^N\}$ se define en términos de la distancia de edición, como la cadena que minimiza el coste combinado de editar M en cada una de las S^i . Siendo $D_c(S^1, \dots, S^N)$ la representación del *coste de edición mínimo* de transformar M en cada una de las S^i , la definición de M es la siguiente:

$$D_c(S^1, \dots, S^N) \equiv \min_{M \in \Sigma^*} \sum_{i=1}^N d_c(M, S^i) \quad (2.9)$$

donde $d_c(M, S^i)$ representa la distancia de edición cíclica entre M y cada S^i .

En general M no es única y el conjunto de todas las cadenas medias posibles se define como:

Definición 2.12 Sea $S = \{S^1, \dots, S^N\}$ el conjunto de cadenas, el conjunto de todas las cadenas medias posibles es

$$Mean(S^1, \dots, S^N) \equiv \{M \in \Sigma^* \mid \sum_{i=1}^N d_c(M, S^i) = D_c(S^1, \dots, S^N)\} \quad (2.10)$$

de donde estamos interesados en encontrar sólo una cadena media representativa $M \in Mean(S^1, \dots, S^N)$.

Como se propone en aproximaciones previas [70, 66], la inferencia de la cadena media se puede llevar a cabo introduciendo el cálculo de D_c en el algoritmo de puesta en correspondencia de cadenas, en nuestro caso cíclicas, descrito en esta sección. Para describir el algoritmo que calcula D_c necesitamos definir δ como, el coste de transformar una subcadena media de un conjunto de subcadenas a cada una de las subcadenas del conjunto:

$$\delta(S_{b_1, e_1}^1, \dots, S_{b_N, e_N}^N) = \min_{X \in \Sigma^*} \sum_{j=1}^N c(X \rightarrow S_{b_j, e_j}^j) \quad (2.11)$$

El algoritmo para calcular D_c , y por tanto M , implica la construcción de una tabla multi-dimensional de distancias T_D , mientras que se construye M . La recurrencia para N cadenas $S^1 = s_1^1 s_2^1 \dots s_{l_1}^1, \dots, S^N = s_1^N s_2^N \dots s_{l_N}^N$ usando la operación de fusión y permitiendo fusionar w caracteres de la cadena S^1 , w de la S^2, \dots , y w de la S^N es:

$$T_D(i_1, \dots, i_N) = \min(T_D(i_1 - r_1, \dots, i_j - r_j, \dots, i_N - r_N) + \delta(S_{i_1 - r_1 + 1, i_1}^1, \dots, S_{i_j - r_j + 1, i_j}^j, \dots, S_{i_N - r_N + 1, i_N}^N)) \quad (2.12)$$

donde $0 \leq r_j \leq w$ y $1 \leq j \leq N$.

T_D se inicializa como sigue:

$$\begin{aligned} T_D(0, \dots, i_j, \dots, i_N) &= 0 \quad \text{para } 0 \leq i_j \leq l_j \\ T_D(0, \dots, i_j, \dots, i_N) &= \infty \quad \text{para } l_j + 1 \leq i_j \leq 2l_j. \end{aligned} \quad (2.13)$$

Entonces $D_c(S^1, \dots, S^N)$ se define como:

$$D_c(S^1, \dots, S^N) = \min(T_D(k_1, \dots, k_j, \dots, k_N)) \quad (2.14)$$

donde $l_j \leq k_j \leq 2l_j - 1$.

2.4.3 Un algoritmo para el cálculo de la cadena media que represente formas poligonales.

Como hemos dicho anteriormente, en el trabajo que presentamos y sin pérdida de generalidad, las formas se aproximan por polígonos. Siguiendo el enfoque introducido por Tsai y Yu [98], una forma se representa por una *cadena de atributos* $S = s_1 \dots s_n$, como en la Fig. 2.4, donde cada segmento de línea s_i se representa por su longitud l_i y su ángulo ϕ_i , el cual se calcula respecto a una línea horizontal de referencia. Los polígonos se comparan usando la distancia de edición de cadenas cíclicas, presentada en la Fig. 2.3, y los costes de las operaciones de edición se definen como una suma ponderada de un coste según el ángulo y según la longitud, de forma parecida a como se hace en [99]. Hemos definido la función de coste de tal modo que una cadena $a_1 a_2$ se fusiona en un símbolo a definiendo los atributos l_a y ϕ_a de la siguiente forma:

$$l_a = l_{\vec{a}_1 + \vec{a}_2} \quad (2.15)$$

$$\phi_a = \phi_{\vec{a}_1 + \vec{a}_2} \quad (2.16)$$

donde \vec{a}_1 es el vector definido por el segmento a_1 y \vec{a}_2 es el vector definido por el segmento a_2 . Una vez se ha definido el símbolo a , se calcula el coste de la fusión siguiendo la siguiente ecuación:

$$c(a_1 a_2 \rightarrow a) = \frac{|\phi_a - \phi_1| l_1}{2\pi l_a} + \frac{|\phi_a - \phi_2| l_2}{2\pi l_a} \quad (2.17)$$

Las ecuaciones previas se pueden extender a la fusión de subcadenas de más de dos símbolos. Se debe hacer notar que las operaciones de sustitución, borrado e inserción son casos particulares de la sustitución en bloque que usa la operación de fusión. Así, el coste total de sustituir toda una secuencia de símbolos por otra ($A_{i,j} \rightarrow B_{k,l}$) se calcula de la siguiente forma:

$$c(A_{i,j} \rightarrow B_{k,l}) = c(A_{i,j}, a) + c(B_{k,l}, b) + c(a, b). \quad (2.18)$$

Siendo a y b los símbolos que se obtienen después de aplicar la operación de fusión $A_{i,j} \rightarrow a$ y $B_{k,l} \rightarrow b$.

El algoritmo del cálculo de la cadena media tiene algunas particularidades, cuando estas cadenas tienen atributos y representan formas 2D aproximadas por polígonos. Para empezar el alfabeto de los símbolos no es finito. Como los símbolos son aristas con sus atributos de longitud l y ángulo ϕ , existe un número infinito de valores posibles. Además no tiene sentido buscar la cadena, en el conjunto de cadenas posibles, que minimice el coste, ya que el conjunto de cadenas posibles también es infinito. En cambio, tiene más sentido *calcular* una cadena que minimice el coste. En nuestro caso, como las formas se aproximan por polígonos y se representan con cadenas con atributos, el problema se transforma en el cálculo de una media geométrica entre polígonos. De manera informal, el algoritmo se podría explicar como sigue:

Dadas dos cadenas A y B , se calcula el algoritmo de puesta en correspondencia, y además, la tabla de distancias T_D . Cada celda $T_D(j, l)$ se asocia con la operación de edición $A_{i,j} \rightarrow B_{k,l}$ que implica la sustitución de una poligonal abierta $A_{i,j}$ de A por una poligonal abierta $B_{k,l}$ de B . La idea es calcular, para cada celda, la

subcadena media C entre $A_{i,j}$ y $B_{k,l}$. En este contexto, el cálculo de la cadena media de un conjunto de cadenas: $S = \{S^1, S^2, \dots, S^N\}$ es equivalente a calcular la forma media M cuyas aristas tengan como valores de atributo los valores medios calculados para las aristas de las cadenas del conjunto S .

El algoritmo sigue la misma estructura básica que el usado para el cálculo de la tabla de distancias T_D , ver ecuación 2.12. Pero necesita una tabla adicional T_M de las mismas dimensiones que T_D para almacenar la subcadena media C que se ha explicado en el párrafo anterior. Así, mientras el algoritmo calcula el coste en una celda $T_D(i_1, \dots, i_N)$, también calcula la subcadena de la cadena media, y la almacena en $T_M(i_1, i_2, \dots, i_N)$. Esta subcadena es la media entre todas las subcadenas S_{b_j, e_j}^j que el algoritmo ha decidido fusionar y substituir en ese punto de la tabla T_D . Al final, para cada celda $T_D(i_1, i_2, \dots, i_N)$, se calcula una $T_M(i_1, i_2, \dots, i_N)$ con el conjunto de aristas de la subcadena media en ese punto. La cadena media se obtiene a partir de las $T_M(i_1, i_2, \dots, i_N)$ asociadas a las celdas involucradas en la secuencia de edición de coste mínimo.

El punto clave en este algoritmo es el paso del cálculo de la subcadena de la cadena media $M(i_1, i_2, \dots, i_N)$ en cada celda de la tabla T_M . Para calcularla se necesita saber qué subcadenas, de las N cadenas que se están comparando, han sido seleccionadas para ser fusionadas y substituidas entre ellas. Luego, si se está calculando la subcadena en $T_M(i_1, i_2, \dots, i_N)$, se ha de saber cual de las operaciones de edición se ha seleccionado en $T_D(i_1, i_2, \dots, i_N)$. Si es la que viene de $T_D(r_1, r_2, \dots, r_N)$, entonces las subcadenas a partir de las cuales se ha de calcular la subcadena media son $S_{r_1, i_1}^1, S_{r_2, i_2}^2, \dots, S_{r_N, i_N}^N$.

El siguiente paso será calcular la cadena media entre las subcadenas, $S_{r_1, i_1}^1, S_{r_2, i_2}^2, \dots, S_{r_N, i_N}^N$. Como una subcadena es en realidad una cadena abierta, y para poder simplificar la notación, las llamaremos cadenas y las denotaremos como S^{r_1}, \dots, S^{r_N} . Luego es necesario calcular su media. La idea es ver las N cadenas como funciones lineales por trozos. Dada una cadena S , denotaremos como F_S la función lineal por trozos inducida por la cadena S . Para calcular la media entre un conjunto de cadenas, calcularemos la función media entre todas las funciones lineales por trozos que las representen. El problema es tener una parametrización común para las funciones lineales, y de este modo poder calcular su media. Para resolver el problema de la parametrización común se han implementado dos métodos. El primero basa la parametrización común en el valor del eje de las x y lo veremos en el apartado 2.4.3. El segundo basa la parametrización común en el porcentaje de longitud de cada arista respecto al total del conjunto, y se explica en el apartado 2.4.3.

Los pasos del algoritmo quedarían como sigue, con una explicación gráfica presentada en la Fig. 2.6, que explica el caso concreto de calcular la media entre dos cadenas A y B , que representan sendas formas.

1. Transformar todos los S^{r_i} en funciones lineales por trozos $F_{S^{r_i}}(x)$, o $F_{S^{r_i}}(lp)$, definidas por la ecuación de la línea de cada segmento, donde x es la posición en el eje de las x y lp es el porcentaje de longitud correspondiente al segmento, ver Fig. 2.6(a).

2. Calcular la media entre las N funciones.

$$\begin{aligned} \text{Método-X: } F_M(x) &= \frac{\sum_{i=1}^N F_{S^i}(x)}{N} \\ \text{Método-L: } F_M(lp) &= \frac{\sum_{i=1}^N F_{S^i}(lp)}{N} \end{aligned} \quad (2.19)$$

Así M será el conjunto de segmentos definidos por $F_M(x)$ o $F_M(lp)$, y el conjunto de segmentos de todas las S^i , ver la Fig. 2.6(g) para el Método-X y la Fig. 2.6(h) para el Método-L.

En las dos subsecciones siguientes vamos a definir los dos métodos para obtener una parametrización común para las funciones por trozos. Luego compararemos ambos métodos y escogeremos uno sobre el que calcularemos su complejidad.

Parametrización común usando el valor del eje x .(Método-X)

Un primer intento para encontrar una parametrización común a las funciones sería usar como parámetro el valor de la coordenada x , $F_{S^i} : \mathbb{R} \rightarrow \mathbb{R}$, pero cuando se trata de hacer esta transformación nos encontramos que en algunos casos obtenemos dos valores y para el mismo valor de x , lo que quiere decir que en estos casos el problema no se puede formular en términos de un cálculo de la media de funciones. Ilustramos este problema en la Fig. 2.5.

Aunque se puede llegar a producir el error anteriormente citado, en la mayoría de los casos esto no se produce y podemos obtener resultados bastante satisfactorios. Por ello hemos hecho un estudio del método y los describimos con mayor detalle a continuación. Las funciones que definirán los segmentos del polígono se definirán en cada una de sus partes utilizando la ecuación del segmento de línea que define la misma. Como las tenemos alineadas, de esta forma ya tienen una parametrización común. En este punto calcular la función media de N funciones es calcular:

$$F_M(x) = \frac{\sum_{i=1}^N F_{S^i}(x)}{N} \quad (2.20)$$

M es la subcadena media de las N cadenas S^1, S^2, \dots, S^N , y está formada por los segmentos de línea definidos por la función lineal por trozos $F_M(x)$.

Desde ahora llamaremos a este método el Método-X.

Parametrización común usando el porcentaje de longitud de cada segmento respecto al total.(Método-L)

Este método basa la parametrización común de las funciones por trozos en el porcentaje de la longitud de cada arista respecto a la longitud total del conjunto de aristas. Así transforma el conjunto de aristas en una función lineal por trozos que toma como parámetro el porcentaje de longitud de cada segmento en relación con la longitud total del conjunto de segmentos $F_{S^i} : [0, 1] \rightarrow \mathbb{R}^2$. Estas funciones están definidas para cada una de sus partes por la ecuación de la línea que representa la

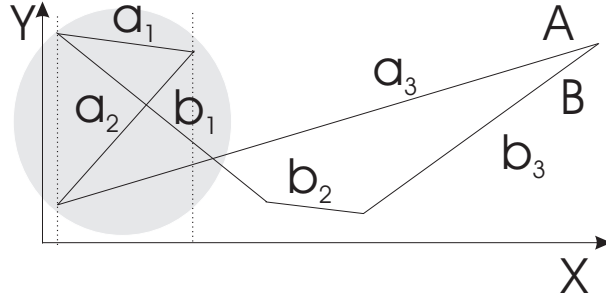


Figura 2.5: El problema que aparece al usar la parametrización común según los valores de x .

arista para ese porcentaje de longitud, y va desde el porcentaje de longitud $[0, 1]$ al (x, y) de la arista que tiene el punto que corresponde a ese porcentaje. Como ese porcentaje va de 0 a 1 en todos los conjuntos de aristas para cada porcentaje, tendremos una correspondencia, es decir, tendremos una parametrización común. Así que para calcular la cadena media entre un conjunto de N cadenas, calcularemos la función media entre las N funciones:

$$F_M(lp) = \frac{\sum_{i=1}^N F_{S'^i}(lp)}{N} \quad (2.21)$$

M es la cadena media de las N cadenas S'^1, S'^2, \dots, S'^N , y está formada por los segmentos de línea definidos por la función lineal por trozos $F_M(lp)$.

Desde ahora llamaremos a este método el Método-L.

2.4.4 Comparación de los dos métodos

Para entender mejor el algoritmo de puesta en correspondencia de dos conjuntos de aristas y su posterior cálculo del conjunto de subcadenas medio usando los dos enfoques explicados en las subsecciones anteriores vamos a ver un ejemplo con dos subcadenas. Este ejemplo se ilustra en la Fig. 2.6. se calcula la cadena media M de dos cadenas A y B usando los dos métodos explicados. Primero se calcula la secuencia de edición n de cadenas de A a B , y luego para cada operación de fusión entre A y B :

1. A través de la distancia de edición de cadenas encontramos la subcadena $A_{i,j}$ que ha sido transformada en la subcadena $B_{k,l}$. Podemos ver la subcadena $A_{i,j}$ en la Fig. 2.6(a) y la subcadena $B_{k,l}$ en la Fig. 2.6(b).
2. Calcular $a'_{i,j}$, la suma vectorial de las aristas de $A_{i,j}$, como podemos ver en la Fig. 2.6(c), y $b'_{k,l}$, la suma vectorial de las aristas de $B_{k,l}$, como podemos ver en la Fig. 2.6(d).
3. A partir de $a'_{i,j}$ y $b'_{k,l}$ calcular m' como $m' = a'_{i,j} \cup b'_{k,l}$, ver la Fig. 2.6(e).

4. Aplicar T_a a $A_{i,j}$ para obtener $A_{i,j}^T$, siendo T_a una transformación afín tal que hace $\vec{m}' = \vec{a}' \cdot T_a$, como podemos ver en la Fig. 2.6(f).
5. Aplicar T_b a $B_{k,l}$ para obtener $B_{k,l}^T$, siendo T_b una transformación afín tal que hace: $\vec{m}' = \vec{b}' \cdot T_b$, como podemos ver en la Fig. 2.6(f).
6. Transformar $A_{i,j}^T$ y $B_{k,l}^T$ en dos funciones lineales por trozos y calcular su media:

(a) Usando el Método-X:

- Transformar $A_{i,j}^T$ en una función lineal por trozos $A_{i,j}^T(x) \rightarrow y$ definida por la ecuación de la línea de cada segmento, ver la Fig. 2.6(g).
- Transformar $B_{k,l}^T$ en una función lineal por trozos $B_{k,l}^T(x) \rightarrow y$ definida por la ecuación de la línea en cada segmento, ver la Fig. 2.6(g).
- Calcular la media entre dos funciones

$$F_M(x) = \frac{F_{A_{i,j}^T}(x) + F_{B_{k,l}^T}(x)}{2}$$

. Luego M será el conjunto de segmentos definidos por $F_M(x)$, y el conjunto medio de los segmentos entre $A_{i,j}$ y $B_{k,l}$, como podemos ver en la Fig. 2.6(g).

(b) Usando el Método-L:

- Transformar $A_{i,j}^T$ en una función lineal por trozos $A_{i,j}^T(lp) \rightarrow (x, y)$ definida por la ecuación de la línea de cada segmento, ver la Fig. 2.6(h).
- Transformar $B_{k,l}^T$ en una función lineal por trozos $B_{k,l}^T(lp) \rightarrow (x, y)$ definida por la ecuación de la línea de cada segmento, ver la Fig. 2.6(h).
- Calcular la media entre las dos funciones

$$F_M(lp) = \frac{F_{A_{i,j}^T}(lp) + F_{B_{k,l}^T}(lp)}{2}$$

. Luego M será el conjunto de segmentos definidos por $F_M(lp)$, y la media del conjunto de segmentos entre $A_{i,j}$ y $B_{k,l}$, como podemos ver en la Fig. 2.6(h). Aquí debemos hacer notar que lp es el porcentaje de longitud. Por ejemplo en la Fig. 2.6(h) las dos primeras partes de segmento alineadas son las dos que tienen el 28% de la longitud total en el conjunto de aristas, las segundas tienen el 12% y así sucesivamente.

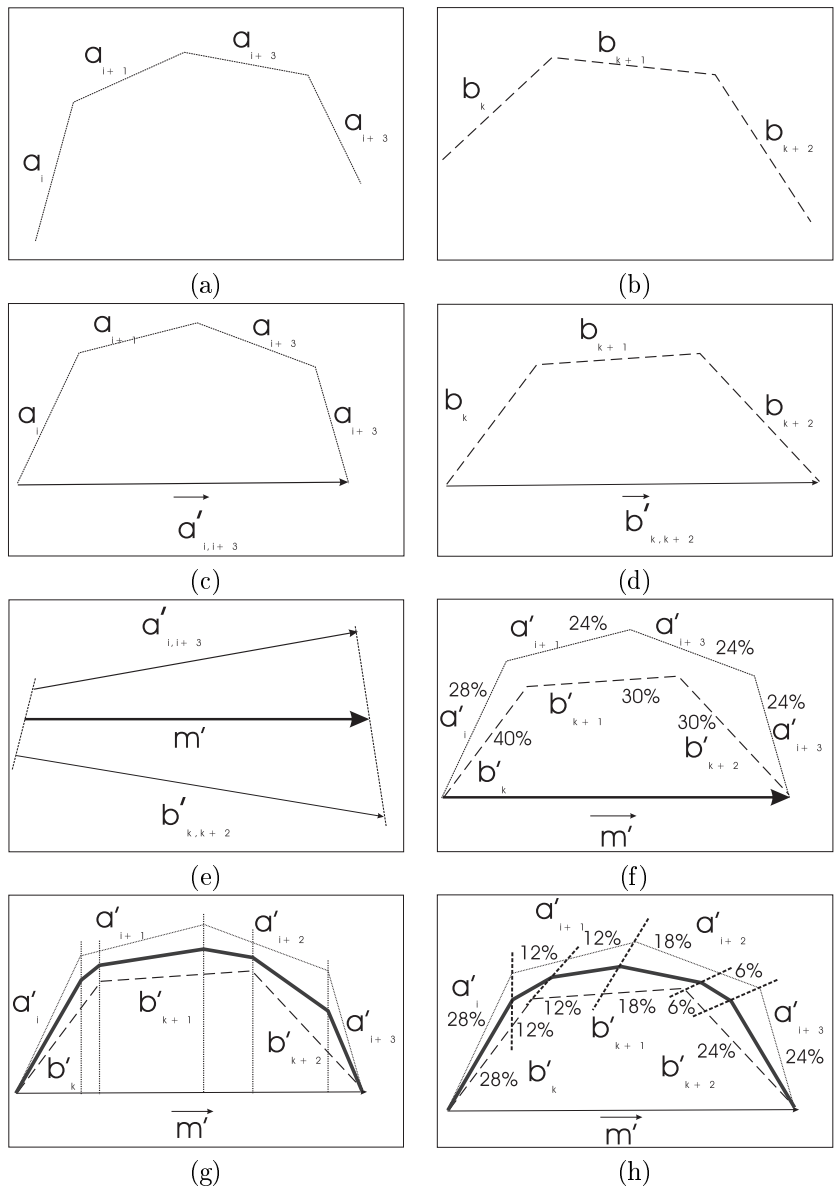


Figura 2.6: Fases del cálculo de la subcadena media M : (a)Conjunto de aristas de $A : A_{i,i+3}$. (b)Conjunto de aristas de $B : B_{k,k+2}$. (c)Suma vectorial del conjunto de aristas de A transformadas en vectores. (d)Suma vectorial del conjunto de aristas de B transformadas en vectores. (e)Cálculo de la media m' entre la suma vectorial de A y B . (f)Alineación de A y B en m' . (g)Subcadena media, en líneas más gruesas, usando el Método-X. (h)Subcadena media, en líneas más gruesas, usando el Método-L.

Hasta este punto se han explicado los dos métodos. También se han presentado los detalles en un paso del cálculo del mismo, y se han comentado paso a paso remarcando las diferencias entre ambos métodos. Ahora vamos a presentar un ejemplo completo, también comentado paso a paso, y representado de forma gráfica en la Fig. 2.7. Se realiza el cálculo de la forma media entre dos polígonos sintéticos usando la distancia de edición de cadenas entre ellos, y en nuestro caso calculado usando el algoritmo que se explica en 2.3.1. A partir de dos polígonos sintéticos mostrados con sus números de arista en las Fig. 2.7(a) el *Pol1* y en la Fig. 2.7(b) el *Pol2*, calculamos la secuencia de edición de cadenas que transforma el *Pol1* en el *Pol2*. La tabla de edición de cadenas que presenta esta secuencia de edición se presenta en la Fig. 2.7(l) y consiste en las siguientes operaciones de edición:

1. Substitución de la arista 0 del *Pol1* por la arista 0 del *Pol2*.
2. Substitución de la arista 1 del *Pol1* por la arista 1 del *Pol2*.
3. Substitución de la arista 2 del *Pol1* por la fusión de las aristas 2 y 3 del *Pol2*.
4. Substitución de la arista 3 del *Pol1* por la fusión de las aristas 4 y 5 del *Pol2*.

Con esta secuencia de edición la forma media se calcula siguiendo los pasos descritos en el apartado 2.4.3, y usando los dos métodos presentados, el Método-X y el Método-L. Estos pasos, en el caso concreto del ejemplo presentado, se muestran en la Fig. 2.7. A partir del *Pol1* y del *Pol2* se seleccionan las aristas y se calculan las aristas medias usando el Método-X y el Método-L, de la siguiente forma:

1. A partir de la arista 0 del *Pol1* y de la arista 0 del *Pol2* calculamos la arista media 0 del polígono media, como puede verse en la Fig. 2.7(e). Este paso es el mismo para ambos métodos.
2. A partir de la arista 1 del *Pol1* y de la arista 1 del *Pol2* se calcula la arista media 1 del polígono media, como puede verse en la Fig. 2.7(f). Este paso es el mismo para ambos métodos.
3. A partir de la arista 2 del *Pol1* y de la arista 2 y la 3 del *Pol2* se calculan las aristas medias 2 y 3 del polígono media. Primero se calcula la suma vectorial de la arista 2 y la 3 del *Pol2* y luego la arista media entre esta suma vectorial y la arista e_2 del *Pol1*, como puede verse en la Fig. 2.7(g). Así lo primero que se hace es alinear la arista 2 del *Pol1* con la suma vectorial de las aristas 2 y 3 del *Pol2*, como podemos ver en la Fig. 2.7(h) y en la Fig. 2.7(i). Hasta este punto el proceso es el mismo para ambos métodos. Luego se calculan la arista 2 y la 3 del polígono media. Las aristas calculadas usando el Método-X se muestran en la Fig. 2.7(h), y las calculadas con el Método-L en la Fig. 2.7(i).
4. A partir de la arista 3 del *Pol1* y las aristas 4 y 5 del *Pol2* se calculan las aristas 4 y 5 del polígono media. Primero se calcula la suma vectorial de las aristas 4 y 5 del *Pol2* y luego la media entre esta suma vectorial y la arista 3 del *Pol1*, como podemos ver en la Fig. 2.7(j). Luego la arista 3 del *Pol1* se alinea con la media calculada, y luego las aristas 4 y 5 del *Pol2* se colocan de forma que

su suma vectorial quede alineada con esa media también, como podemos ver en la Fig. 2.7(k). Hasta este punto el proceso es el mismo para ambos métodos. Luego se calculan las aristas 4 y 5 del polígono media. En este caso especial las aristas calculadas usando el Método-X y las resultantes de usar el Método-L se muestran en la Fig. 2.7(k), y son ambas equivalentes, dado que las aristas 4 y 5 del *Pol2* tienen cada una el 50% de la longitud total.

Al final tenemos las dos formas medias construidas a partir de las aristas medias calculadas. La Fig. 2.7(c) muestra la forma media que resulta usando el Método-X, mientras que la Fig. 2.7(d) muestra la forma media calculada con el Método-L.

Como vemos en estos ejemplos ambos métodos funcionan bastante bien pero el Método-X tiene el problema descrito anteriormente por ello seleccionamos el Método-L para el cálculo de la forma media. En la Fig. 2.8 se muestra el problema del Método-X en el cálculo de la forma media entre dos polígonos, donde aparece un segmento incorrecto. La Fig. 2.8(a) muestra el *pol1*, la Fig. 2.8(b) el *pol2* y la Fig. 2.8(c) el polígono media calculado. La Fig. 2.8(d) muestra la tabla de edición de cadenas y a partir de ella vemos que las aristas 2 y 3 de un polígono se fusionan y substituyen por la fusión de las aristas 3, 4 y 5 del otro. En la Fig. 2.8(e) podemos ver la alineación de estas 5 aristas y podemos ver que la arista 3 y la 4 del *subpol2* no pueden ser transformadas en una función por trozos ya que algunas x tienen múltiples imágenes. Esto hace que el algoritmo, usando el Método-X no funcione correctamente en este caso. Pero usando el Método-L podemos ver el resultado correcto en la Fig. 2.8(f).

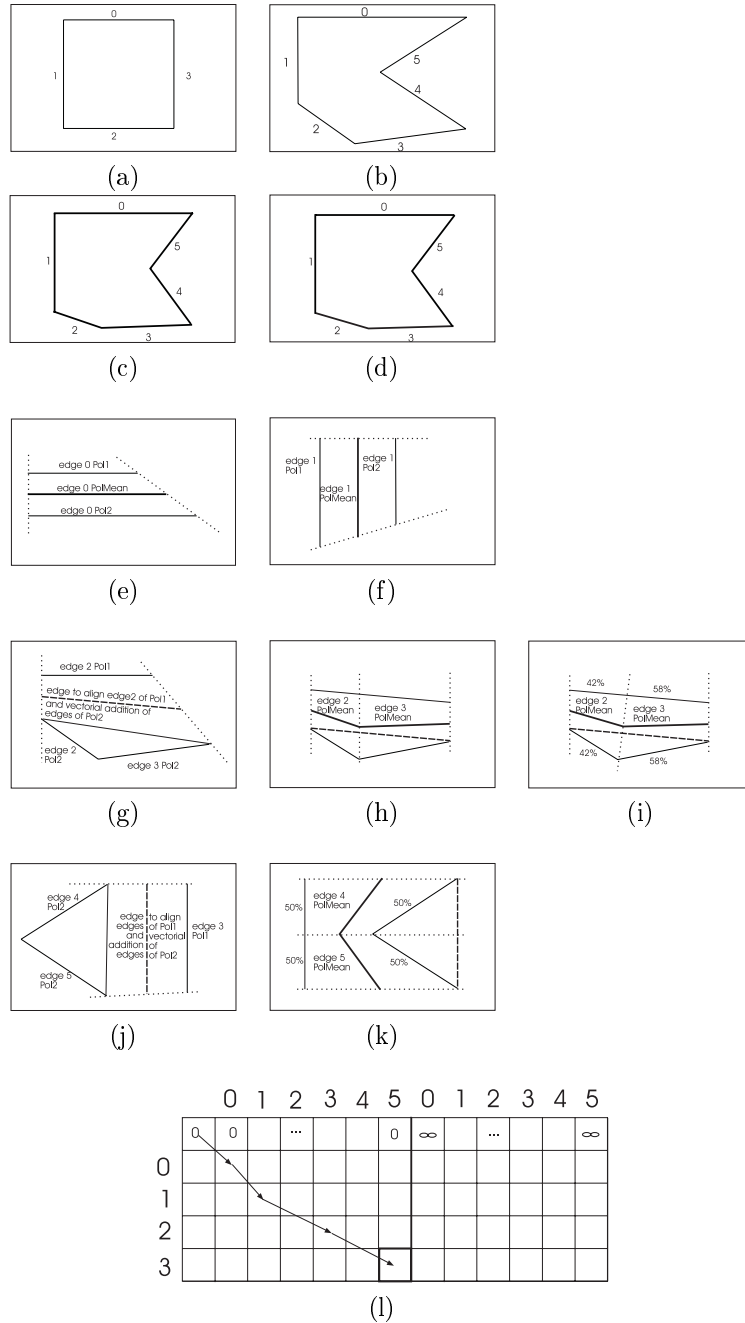


Figura 2.7: Pasos del algoritmo del cálculo de la forma media a partir de dos formas sintéticas. (a) *Pol1*. (b) *Pol2*. (c) Polígono media *Pol1-Pol2* usando el Método-X. (d) Polígono media *Pol1- Pol2* usando el Método-L. (e) Arista media *ed0-ed0* (para Método-X y Método-L). (f) Arista media *ed1-ed1*, (para Método-X y Método-L). (g) Alineación de las aristas del *Pol1* y el *Pol2*. (h) Aristas medias *ed2-Fusión(ed2-ed3)* usando el Método-X. (i) Aristas medias *ed2-Fusión(ed2-ed3)* usando el Método-L. (j) Alineación de las aristas del *Pol1* y el *Pol2*. (k) Aristas medias *3-Fusión(4-5)* usando ambos métodos. (l) Tabla de la distancia de edición de cadenas entre el *Pol1* y el *Pol2*.

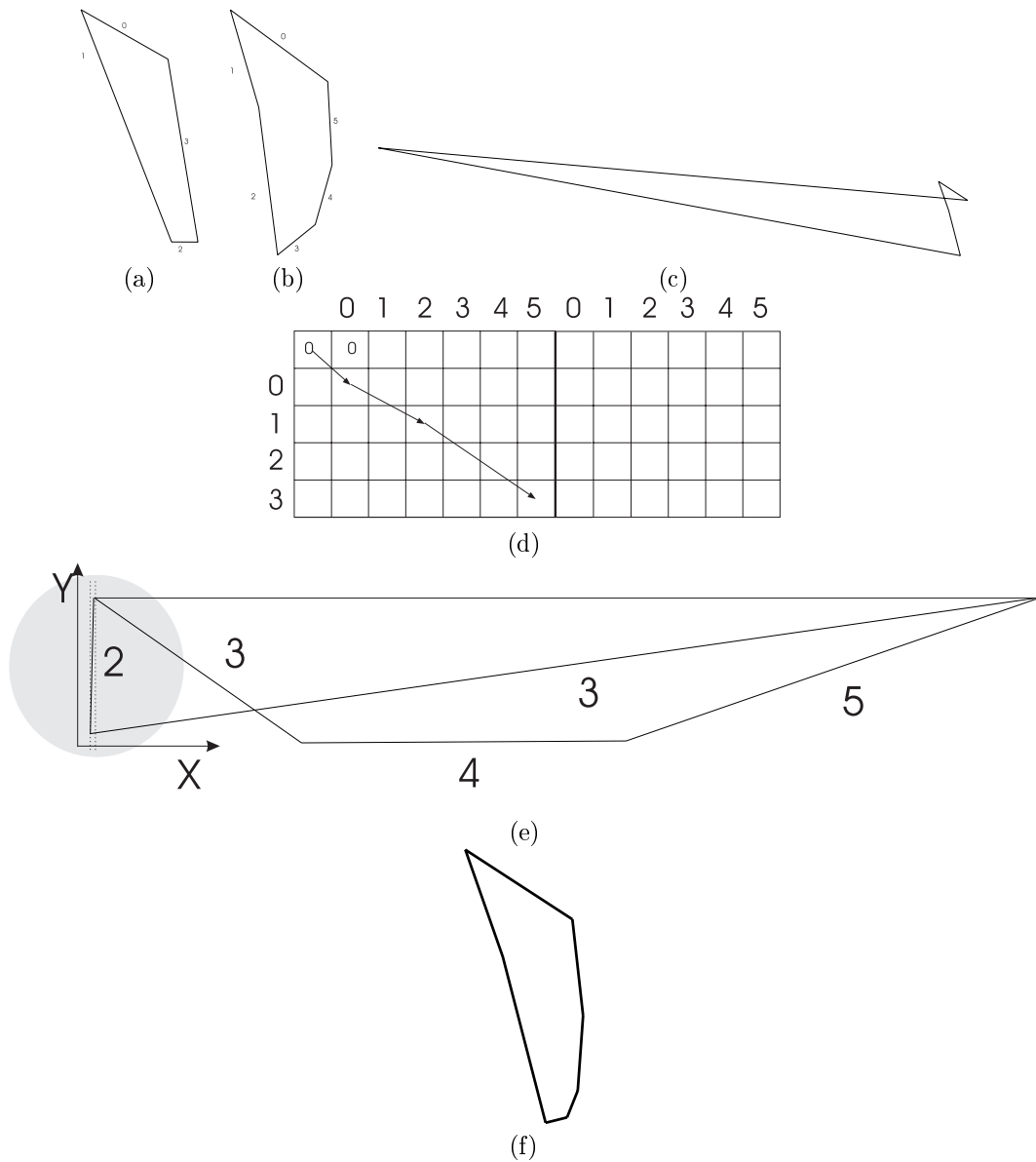


Figura 2.8: el problema del Método-X en detalle. (a)*Pol1*. (b)*Pol2*. (c)Forma media entre *Pol1* y *Pol2* usando el Método-X. (d)Tabla de distancia de edición entre *Pol1* y *Pol2*. (e)Detalle de la alineación de las aristas 2,3 y 4. (f)Forma media entre *Pol1* y *Pol2* usando el Método-L.

2.4.5 Análisis de la complejidad

El problema del cálculo de la forma media a partir de un conjunto de formas tiene un coste computacional elevado, ya que es necesario calcular dos tablas T_D y T_M , y a partir de la ecuación 2.12 se puede deducir que el cálculo de la tabla T_D es un problema exponencial respecto al número de cadenas. En general el cálculo de la cadena media es un problema NP-completo como se prueba en [30]. Dada esta alta complejidad, hemos realizado algunos cambios en el algoritmo para hacerlo útil. Para ello, en vez de calcular las tablas T_D y T_M para N formas, se calcula la forma media entre ellas de una manera progresiva, es decir, calculándola entre dos y añadiendo una forma más en cada paso. Esta modificación hace que tengamos una solución subóptima, dado que la puesta en correspondencia entre las aristas que forman las formas no será siempre exactamente la misma cuando la calculamos entre todas las formas al mismo tiempo que cuando la calculamos forma a forma. El problema de esta adaptación es garantizar que cada forma tiene el mismo peso en el proceso. Por ello se necesita mantener guardado, para cada forma, el número de formas a las que representa, y la suma parcial de las funciones lineales por trozos que la describen. De esta forma podemos asegurar que cuando dos formas medias se agrupen, la que represente más formas tendrá más peso al calcular la media entre ellas. Así cada arista tendrá como atributos, además de la longitud l y el ángulo ϕ , la suma parcial de todas las aristas que representa, tratadas como funciones lineales por trozos, y el número de aristas que representa. Con todos estos atributos se puede mantener la suma parcial en cada paso, y calcular la cadena media de un conjunto en cada paso del algoritmo. Este comportamiento permite también paralelizar el cálculo de la forma media de un conjunto grande de formas. Un ejemplo de esta manera incremental de calcular la forma media se presenta en la Fig. 2.9, donde podemos ver las formas S^1 , S^2 , S^3 , S^4 y S^5 y cómo se puede calcular su media. Primero, se calculan las medias $S^{1,2}$ y $S^{3,4}$. Luego, la forma media $S^{1,\dots,4}$, para finalizar añadiendo una sola forma S^5 a la media calculada hasta ese momento, obteniendo la media $S^{1,\dots,5}$. Podemos ver que S^5 tiene menos peso que $S^{1,\dots,4}$, ya que sólo representa una forma, mientras que $S^{1,\dots,4}$ representa cuatro. La forma tachada sería la media entre $S^{1,\dots,4}$ y S^5 si ambas representaran el mismo número de formas.

Hasta este punto hemos explicado como reducir el problema mediante una aproximación: en vez de calcular una gran T_D , calculamos varias T'_D s menores, una por cada comparación entre dos formas. Pero también se puede reducir el número de operaciones necesarias para calcular T_M . de hecho, no es necesario calcular toda la tabla T_M . Sería suficiente con calcular primero la tabla de distancias T_D , y luego la cadena media asociada a la secuencia de edición subyacente. Así solamente se calcularían las subcadenas necesarias para la cadena media, y no se realizarían otros cálculos inútiles. Para realizar el análisis de la complejidad computacional del algoritmo deberíamos realizar las siguientes consideraciones. Siendo N el número de polígonos y n su número de aristas. Primero el coste mínimo de la secuencia de edición de cadenas entre dos formas codificadas como cadenas se calcula en orden $O(N^2 \log n)$ ver [68]. Segundo, la cadena media entre dos cadenas, siguiendo el método descrito en el Apartado 2.4.3, se puede calcular en orden $O(n)$. Así, el cálculo incremental de la media de N cadenas requiere en el peor de los casos $O(Nn^2 \log n)$. Nuestro procedimiento subóptimo para

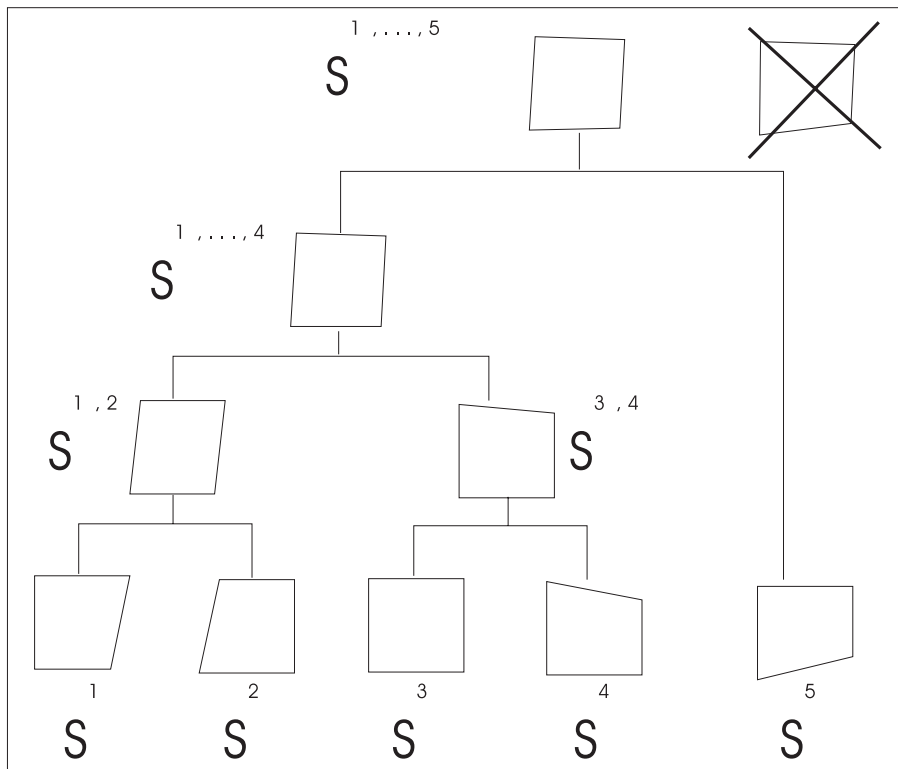


Figura 2.9: Método progresivo para calcular la forma media. En el último paso podemos ver que la forma $\overline{S^{1234}}$ tiene más influencia que la forma S_5 . La figura tachada con una cruz sería la forma media obtenida si S_5 y $\overline{S^{1234}}$ tuvieran el mismo peso.

calcular la cadena media tiene una complejidad incluso menor que la de calcular la cadena mediana de conjunto, cuya complejidad es $O(N^2n^2 \log n)$.

2.5 Resultados

En esta sección se presentan dos tipos de experimentos. En el primero el algoritmo que calcula la cadena media se aplica a formas sintéticas. En el segundo se aplica a aproximaciones poligonales de formas reales obtenidas de la base de datos Rutgers tools¹. Ambos experimentos se realizan cogiendo las formas dos a dos y comparando los resultados del Método-X con los del Método-L. Para finalizar, mostramos algunos resultados del cálculo de la forma media de una manera progresiva, de un grupo de imágenes sintéticas y un grupo de aproximaciones poligonales de formas reales. Los resultados específicos del entorno en el que hemos aplicado este trabajo se presentan en el capítulo 5.

2.5.1 Resultados del cálculo de la forma media entre dos formas sintéticas

En este apartado mostramos como funciona el algoritmo con polígonos sintéticos. La Tabla 2.1 muestra formas medias entre ocho formas sintéticas usando el Método-X. Las formas medias que aparecen en cada celda (i, j) son el resultado de calcular la forma media entre las formas que aparecen al principio de la fila j y la columna i . Debemos notar que las formas medias entre un polígono y él mismo, que aparecen en la diagonal, han sido calculadas y corresponden con la propia forma. Las formas medias que aparecen son, en general, las esperadas, según la inclinación y partición de sus aristas. Pero en algunos casos los resultados son asimétricos, por ejemplo la forma media entre *hexa* y *halfst2* no es exactamente la misma que la calculada entre *halfst2* y *hexa*. Esto es debido a que la alineación que hacemos a partir de la comparación usando la distancia de edición tampoco es simétrica y nos puede dar alineaciones diferentes si comparamos dos polígonos en una dirección o en la contraria.

La Tabla 2.2 muestra otro conjunto de formas medias entre las mismas ocho formas sintéticas, pero usando el Método-L. La manera de mostrar los ejemplos es la misma que en el ejemplo previo. El problema con las asimetrías aparece como en la tabla anterior y los resultados de las formas medias son casi iguales. Así que podemos concluir que para este tipo de formas, donde no aparece el problema mostrado en la Fig. 2.8, ambos métodos funcionan de manera bastante similar.

¹Esta base de datos puede encontrarse en: <http://www.cs.rutgers.edu/pub/sven/rutgers-tools/>


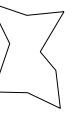

b2									
b1									
st2									
halfst2									
cuad2									
pent									
hexa									
octo									
									
		octo	hexa	pent	cuad2	halfst2	st2	b1	b2

Tabla 2.1: Cadena media entre 8 imágenes sintéticas, cogidas de dos en dos, usando el Método-X.


























































































Tabla 2.2: Cadena media entre 8 imágenes sintéticas, cogidas de dos en dos, usando el Método-L.

2.5.2 Resultados de calcular la forma media entre dos formas reales.

En este apartado presentamos algunos resultados de aplicar el algoritmo a aproximaciones poligonales de formas reales. Estas formas reales han sido extraídas de la base de datos Rutgers tools. Los resultados de calcular la forma media entre diez imágenes cogidas de dos en dos usando el Método-X se presentan en la Tabla 2.3, mientras que los resultados sobre las mismas imágenes pero utilizando el Método-L se presentan en la Tabla 2.4. Las formas medias que se muestran en la tabla han sido calculadas de la siguiente manera, para cada columna se calcula la forma media entre el polígono que figura al principio de la columna y el que está en la fila correspondiente. Debemos hacer notar que las formas medias presentadas como resultado de calcular la media entre una forma y ella misma han sido calculadas igual que el resto. Estas imágenes tienen entre treinta y sesenta aristas y muchas veces su contorno no es suave sino abrupto, todo ello incrementa la dificultad del cálculo de la forma media.

En estas tablas podemos ver que el algoritmo funciona bastante bien pero tiene un problema que se presenta en ambos métodos. El problema común que tiene ha sido ya explicado para las formas sintéticas. A veces el cálculo de la forma media entre dos no es simétrico, y debido a ello tenemos resultados diferentes si la calculamos en un sentido o en el contrario. Esto pasa porque la distancia de cadenas no es simétrica y nos da una alineación diferente de aristas, como hemos dicho en la Subsección 2.5.1. El segundo problema sólo aparece en la tabla que presenta el Método-X y ya fue presentado en la Fig. 2.5 y discutido en la Subsección 2.6. Podemos observar este problema en el cálculo de la forma media entre *Pl3* y *Plso41*, y en el cálculo entre *ori1* y *Pl3*. La Fig. 2.10 nos muestra las dos formas medias incorrectas calculadas en detalle. Podemos ver que la forma media se calcula correctamente hasta un cierto punto en el que hay una arista que va en un sentido incorrecto y a partir de ese punto la forma continua correctamente. A parte de en estos casos, el resto presenta las formas esperadas visualmente para los dos casos. Concretamente si miramos las características de la curvatura de sus mangos y la abertura de sus pinzas. Veamos, por ejemplo, la forma media calculada entre el *Pl6* y el *Pl5*, dos polígonos notablemente diferentes en cuanto a las características mencionadas. La media obtenida, tanto con el Método-X como con el Método-L, presenta una abertura de las pinzas menor que la del *Pl6* y mayor que la del *Pl5*, lo mismo pasa con la curvatura de sus mangos, que es más redondeada que la de *Pl6* pero menos que la de *Pl5*. Debemos remarcar que el Método-L funciona correctamente con todas las imágenes y nos da resultados iguales o mejores que el Método-X para todas las formas medias. Por ejemplo podemos ver resultados mejores en las medias calculadas entre *Pl6* y *pl8* y entre *pl7* y *pl3*, y resultados sin aristas incorrectas en las formas medias calculadas entre *ori1* y *pl3*, y entre *pl3* y *so41*.

Tabla 2.3: Forma media de 10 formas reales, cogidas de dos en dos, de la base de datos Rutgers tools usando el Método-X.

		pl6	plso41	plso42	pl5	ori1	ori3	pl1	P13	P17	P18
											
pl6											
so41											
so42											
pl5											
ori1											
ori3											
pl1											

64 CORRESPONDENCIA DE FORMAS 2D Y CÁLCULO DE LA FORMA MEDIA

Tabla 2.3: Forma media de 10 formas reales, cogidas de dos en dos, de la base de datos Rutgers tools usando el Método-X.












































	p16	plso41	plso42	p15	ori1	ori3	p11	P13	P17	P18	
											
p13											
p17											
p18											

Tabla 2.4: Forma media de 10 formas reales, cogidas de dos en dos, de la base de datos Rutgers tools usando el Método-L.
































































































































		P16	so41	so42	pl5	ori1	ori3	pl1	P13	P17	P18
											
P16											
so41											
so42											
P15											
ori1											
ori3											
pl1											

Tabla 2.4: Forma media de 10 formas reales, cogidas de dos en dos, de la base de datos Rutgers tools usando el Método-L.

	P16	so41	so42	pl5	ori1	ori3	pl1	P13	P17	P18
										
P13										
P17										
P18										

2.5.3 Resultados de calcular la forma media entre un grupo de formas.

En este apartado evaluamos el comportamiento del algoritmo que calcula la forma media entre un grupo de formas, primero calculando la media entre dos formas y luego añadiendo en cada paso una nueva forma a la media calculada. Como en los resultados de calcular la media dos a dos calculamos la forma media usando el Método-X y el Método-L y comparando sus comportamientos.

El primer grupo son formas sintéticas a partir de las que hemos calculado su forma media. En la Fig. 2.11 se presentan los resultados usando el Método-X, mientras que la Fig. 2.12 muestra los resultados usando el Método-L. Para el Método-X podemos ver en la Fig. 2.11(a), Fig. 2.11(b), Fig. 2.11(c) y Fig. 2.11(d) las cuatro formas a partir de las que se calcula la forma media. La Fig. 2.11(g) es la forma media de todas ellas calculada usando el Método-X. La Tabla 2.5 muestra la distancia de edición de cadenas entre todas las formas cogidas dos a dos, incluyendo la forma media calculada. También aparece el promedio de las distancias de cada forma. Para el Método-L podemos ver en la Fig. 2.12(a), la Fig. 2.12(b), la Fig. 2.12(c) y la Fig. 2.12(d) las cuatro formas a partir de las cuales vamos a calcular su forma media. La Fig. 2.12(g) es la forma media entre todas ellas usando el Método-L. En la Tabla 2.6 se muestra la distancia de edición de cadenas entre cada forma y todas las demás, incluyendo la

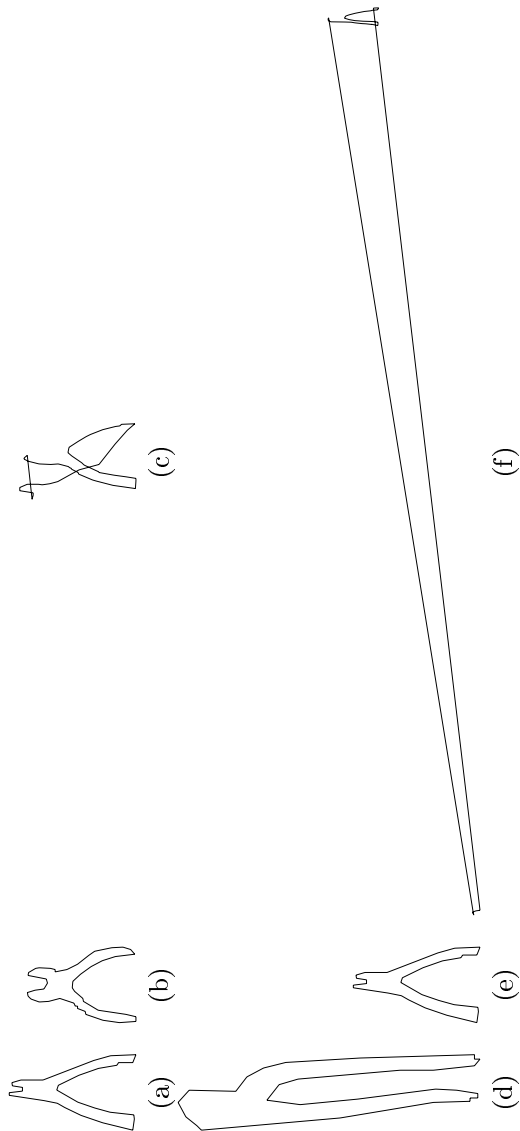


Figura 2.10: Dos ejemplos del mismo problema calculando la forma media de 2 formas reales de la base de datos Rutgers tools. (a) *ori1*. (b) *so41*. (c) Media entre *ori1* y *so41* usando el Método-X. (d) *ori1*. (e) *pl3*. (f) Media entre *ori1* y *Pl3* usando el Método-X.

forma media calculada, también aparece el promedio de estas distancias para cada forma. Vemos como tanto en el caso del Método-X como en el del Método-L las formas medias acumuladas van reflejando las formas que representan, por ejemplo, tanto en la Fig. 2.11(e) como en la Fig. 2.12(e) vemos como la punta de estrella de la parte inferior izquierda queda marcada de forma más tenue que el de una de las dos formas que representa y más marcada que en la otra, donde es inexistente. De la misma manera vemos que las medias acumuladas en la Fig. 2.11(f) y en la Fig. 2.12(f) presentan ahora las cuatro puntas de estrella atenuadas por la influencia del cuadrado y la figura con solo una punta, y como de las cuatro puntas la que aparece en dos de las figuras está más marcada, es decir la inferior derecha.

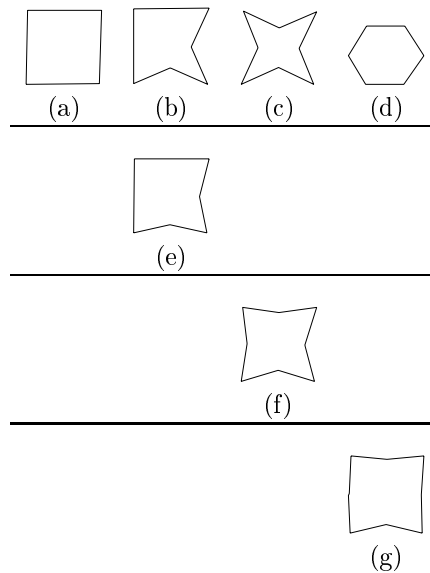


Figura 2.11: Cadena media de 3 imágenes sintéticas calculada según el Método-X. (a)*cuad2*. (b)*halfst2*. (c)*st2*. (d)*hexa*. (e)Polígono media entre *cuad2* y *halfst2*. (f)Polígono media acumulado entre Polígono en (e) y *st2*. (g)Polígono media acumulado entre el Polígono en (f) y *hexa*.

	cuad2	halfst2	st2	hexa	Mean Shape
cuad2	0	0,353953	0,686108	0,625331	0,175139
halfs2	0,353953	0	0,379438	0,914431	0,390360
st2	0,686108	0,379438	0	1,389160	0,721279
hexa	0,625331	0,914431	1,389160	0	0,727951
Valor promedio	0,4163480	0,4119555	0,61367649	0,73223049	0,50368225

Tabla 2.5: Tabla de distancias entre las poligonales cuad2, halfst2, st2, hexa y la forma media que las representa calculada usando el Método-X.

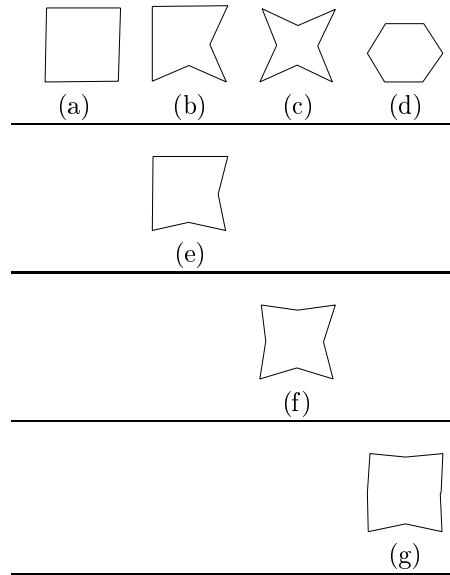


Figura 2.12: Cadena media de 3 imágenes sintéticas calculada según el Método-L. (a)*cuad2*. (b)*halfst2*. (c)*st2*. (d)*hexa*. (e)Polígono media entre *cuad2* y *halfst2*. (f)Polígono media acumulado entre Polígono en (e) y *st2*. (g)Polígono media acumulado entre el Polígono en (f) y *hexa* .

	cuad2	halfst2	st2	hexa	Mean Shape
cuad2	0	0,353953	0,686108	0,625331	0,178412
halfst2	0,353953	0	0,379438	0,914431	0,384863
st2	0,686108	0,379438	0	1,389160	0,721034
hexa	0,625331	0,914431	1,389160	0	0,721565
MeanValue	0,4163480	0,4119555	0,61367649	0,73223049	0,5014685

Tabla 2.6: Tabla de distancias entre las poligonales cuad2, halfst2, st2, hexa y la forma media que las representa calculada usando el Método-L.

El segundo y tercer grupo de formas se han extraído de la base de datos de Rutgers tools y a partir de ellas se calcula la media. En la Fig. 2.13 usando el Método-X mientras que en la Fig. 2.14 usando el Método-L. Para este segundo grupo podemos ver en la Fig. 2.13(a), la Fig. 2.13(b), la Fig. 2.13(c) y la Fig. 2.13(d) las formas, y además en la Fig. 2.14(a), la Fig. 2.14(b), la Fig. 2.14(c), la Fig. 2.14(d) las mismas formas. La Fig. 2.13(g) presenta la forma media de todo el grupo calculada usando el Método-X, y la Fig. 2.14(g) también muestra la media de todas ellas pero calculada usando el Método-L. La Tabla 2.7 muestra las distancias de edición entre cada forma y todas las demás, incluida la forma media calculada según el Método-X. Mientras que la Tabla 2.8 muestra la misma información pero usando la forma media calculada según el Método-L. El tercer grupo de formas y su forma media se presentan de la misma manera en la Fig. 2.15 y la Fig. 2.16. Podemos ver el grupo de imágenes en la Fig. 2.15(a), la Fig. 2.15(b), la Fig. 2.15(c) y la Fig. 2.15(d), así como en la Fig. 2.16(a), la Fig. 2.16(b), la Fig. 2.16(c) y la Fig. 2.16(d). Sus formas medias se presentan en la Fig. 2.15(g) usando el Método-X y en la Fig. 2.16(g) usando el Método-L. Las tablas con las distancias de edición y las distancias promedio se presentan en la Tabla 2.9 para el Método-X y en la Tabla 2.10 usando el Método-L. En todos los casos vemos como la forma media se va modificando a medida que añadimos una nueva forma a representar y que esto lo podemos ver en el redondeado de sus mangos, y la abertura de sus pinzas, así como en el tamaño total de la forma .

	pl5	so41	so42	pl6	Mean Shape
pl5	0	4,61569	3,52772	3,62336	3,427922
so41	4,61569	0	2,30095	2,94666	2,963549
so42	3,52772	2,30095	0	1,54205	2,660026
pl6	3,62336	2,94666	1,54205	0	2,232773
MeanValue	2,94169250	2,46582500	1,84268000	2,02801749	2,8210675

Tabla 2.7: Tabla de distancias entre las poligonales *Pl5*, *so42*, *so41*, *Pl6* y su forma media correspondiente, calculada según el Método-X.

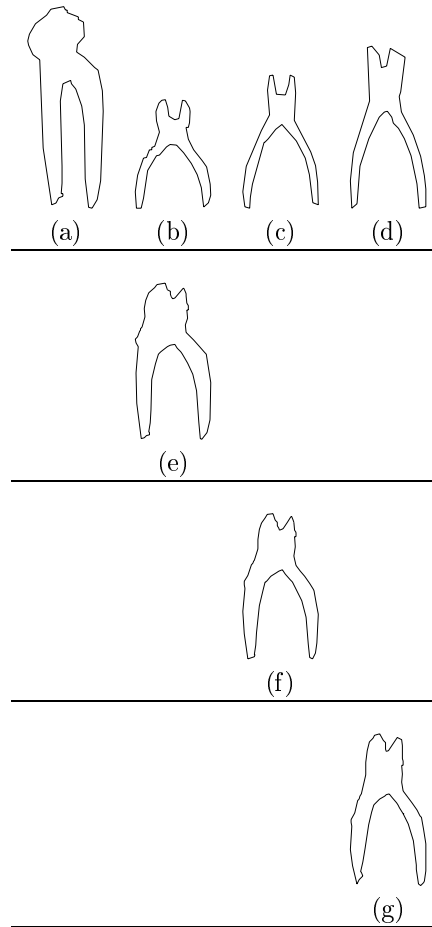


Figura 2.13: Cadena media de 3 formas reales obtenidas de la base de datos de Rutgers, usando el Método-X. (a) *PI5*. (b) *so41*. (c) *so42*. (d) *PI6*., (e) Polígono media entre *PI5* y *so41*. (f) Polígono media entre el polígono media acumulado en (e) y *so42*. (g) Polígono media entre el Polígono media acumulado en (f) ay *PI6*.

74CORRESPONDENCIA DE FORMAS 2D Y CÁLCULO DE LA FORMA MEDIA

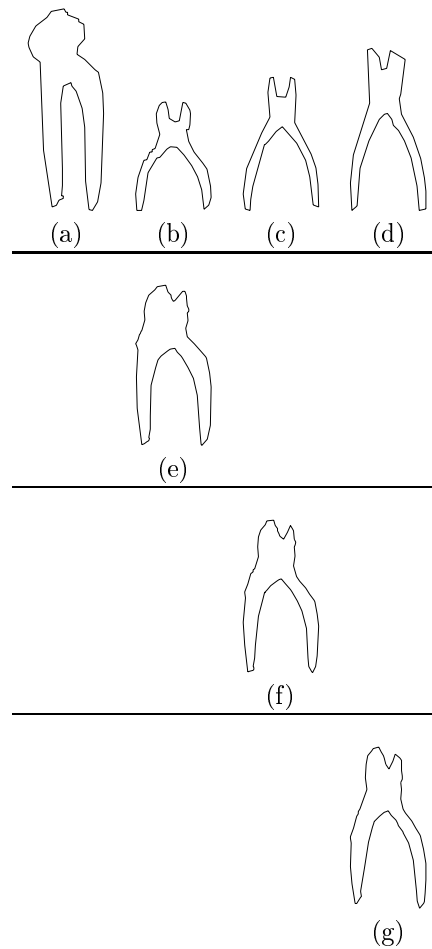


Figura 2.14: Cadena media de 3 formas reales obtenidas de la base de datos de Rutgers, usando el Método-L. (a)*P15*. (b)*so41*. (c)*so42*. (d)*P16*., (e)Polígono media entre *P15* y *so41*. (f)Polígono media entre el polígono media acumulado en (e) y *so42*. (g)Polígono media entre el Polígono media acumulado en (f) ay *P16*.

	pl5	so41	so42	pl6	MeanShape
pl5	0	4,61569	3,52772	3,62336	3,547381
so41	4,61569	0	2,30095	2,94666	2,41876
so42	3,52772	2,30095	0	1,54205	2,38927
pl6	3,62336	2,94666	1,54205	0	2,15026
MeanValue	2,94169250	2,46582500	1,84268000	2,02801749975	2,62641775

Tabla 2.8: Tabla de distancias entre las poligonales *Pl5*, *so42*, *so41*, *Pl6* y su forma media correspondiente, calculada según el Método-L.

76 CORRESPONDENCIA DE FORMAS 2D Y CÁLCULO DE LA FORMA MEDIA

	ori3	pl6	so41	so42	Mean Shape
ori3	0	2,06748	3,29746	2,18126	2,06149
pl6	2,06748	0	2,94666	1,54205	1,75586
so41	3,29746	2,94666	0	2,30095	2,28058
so42	2,18126	1,54205	2,30095	0	1,58191
MeanValue	1,88655	1,63904750025	2,13626750025	1,50606500025	1,91996

Tabla 2.9: Tabla de distancias de edición entre todas las formas *ori3*, *pl6*, *so42*, *so41* y su forma media calculada usando el Método-X.

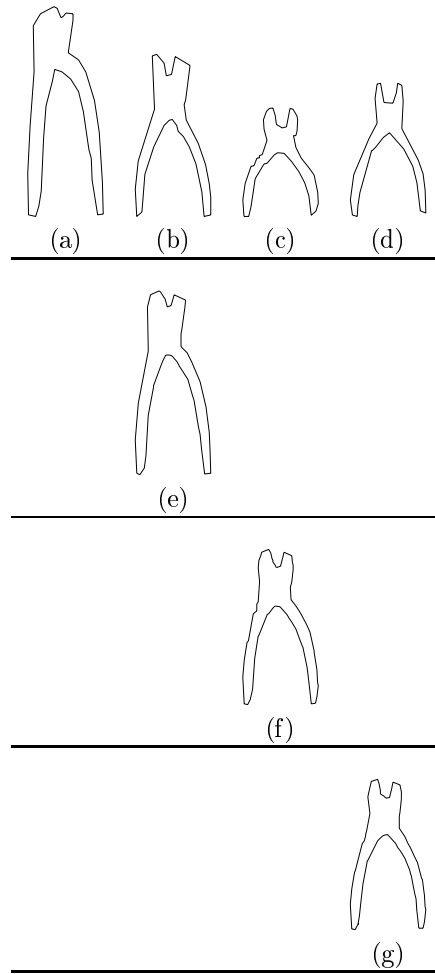


Figura 2.15: Cadena media de 3 formas reales obtenidas de la base de datos de Rutgers tools, calculada usando el Método-X. (a)*ori3*. (b)*pl6*. (c)*so41*. (d)*so42*. (e)Polígono media entre *ori3* y *pl6*. (f)Polígono media entre el Polígono acumulado en (e) y *so41*. (g)Polígono media entre el Polígono media acumulado en (f) y *so42*.

78CORRESPONDENCIA DE FORMAS 2D Y CÁLCULO DE LA FORMA MEDIA

	ori3	pl6	so41	so42	Mean Shape
ori3	0	2,06748	3,29746	2,18126	2,00879
pl6	2,06748	0	2,94666	1,54205	1,48301
so41	3,29746	2,94666	0	2,30095	2,36735
so42	2,18126	1,54205	2,30095	0	1,54164
MeanValue	1,88655	1,63904750025	2,13626750025	1,50606500025	1,8501975

Tabla 2.10: Tabla de distancias de edición entre todas las formas *ori3*, *pl6*, *so42*, *so41* y su forma media calculada usando el Método-L.

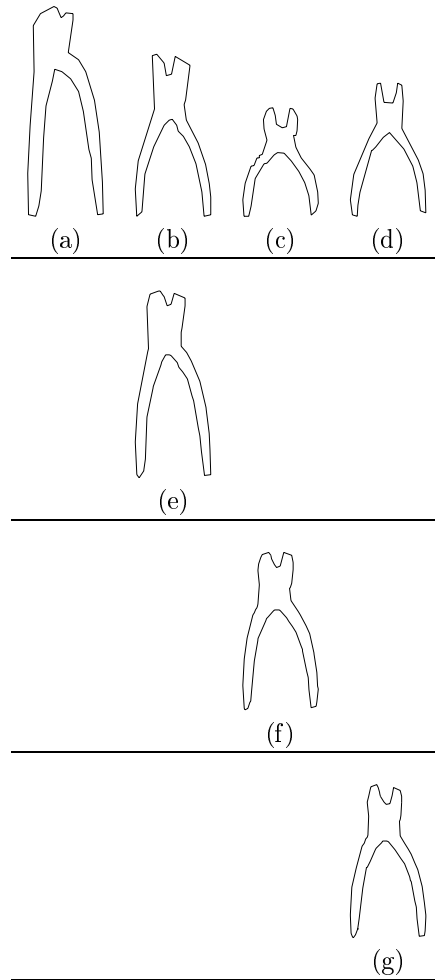


Figura 2.16: Cadena media de 3 formas reales obtenidas de la base de datos de Rutgers tools, calculada usando el Método-L. (a)*ori3*. (b)*pl6*. (c)*so41*. (d)*so42*. (e)Polígono media entre *ori3* y *pl6*. (f)Polígono media entre el Polígono acumulado en (e) y *so41*. (g)Polígono media entre el Polígono media acumulado en (f) y *so42*.

Debemos observar que la forma media obtenida para un grupo de formas dado tiene un elevado número de aristas. Esto es debido a que se produce una fragmentación en las aristas de la misma. Cuando se calcula la forma media de forma progresiva, es decir, añadiendo más y más formas para ser representadas, estamos añadiendo de hecho una forma nueva en cada paso, pero también hacemos que se fragmenten las aristas de la forma media cada vez más. Esto puede producir un incremento en las distancias de edición de cadenas de la forma media a todas las demás. Por ejemplo en la Tabla 2.5 y en la Tabla 2.6 podemos ver que la forma que tiene la distancia promedio menor es la *halfst2*. En la Tabla 2.7 y la Tabla 2.8 es la forma *pl6*. En la Tabla 2.9 y la Tabla 2.10 es la forma *so42*. Esto es debido a que al comparar dos formas con un grado diferente de partición de sus aristas, la que tiene las aristas más particionadas debe fusionar las mismas, y la operación de fusión tiene un coste asociado. De hecho es un problema de “resolución”, ya que la forma media se presenta con una resolución mayor, o un nivel de detalle superior al de las formas originales, y por ello sus distancias no son comparables. Para resolver este problema deberíamos modificar el coste de la operación de fusión, en los casos en que la fragmentación de la forma media lo necesitase. O de forma alternativa, deberíamos transformar la forma media hasta que tuviera la misma resolución que las formas a las que representa.

2.6 Conclusiones

En este capítulo hemos presentado el problema de la correspondencia de formas. Hemos visto cómo una forma puede representarse por medio de una cadena y cómo estas cadenas pueden aumentar su poder de representación añadiéndoles atributos. También hemos visto cómo comparar cadenas usando operaciones de edición para transformar una cadena en otra y cómo estas operaciones han de tener asociado un coste que al irlo acumulando en cada una de las operaciones de edición de una cadena a otra nos dará la distancia entre las mismas. Hemos comprobado cómo este proceso necesita una modificación cuando las cadenas representan formas poligonales cerradas ya que necesitamos saber por dónde empezar a comparar las mismas. Para solucionar este problema normalmente se consideran cadenas cíclicas y se modifica el algoritmo de comparación entre ellas.

La aplicación en la que se enmarca esta tesis exige no sólo la definición de un criterio de semejanza entre formas 2D, resuelto mediante la distancia de edición de cadenas, sino también la necesidad de un algoritmo que, en base al mismo modelo de representación, infiera la forma media representativa de un conjunto dado. Así hemos presentado un nuevo algoritmo sub-óptimo para el cálculo de la forma media de un grupo de formas dado. Como las formas se representan mediante cadenas cíclicas, con atributos que representan sus características geométricas, el cálculo de la forma media se formula en términos del cálculo de la cadena cíclica media. La idea clave del algoritmo es transformar las cadenas que representan las formas en funciones lineales por trozos, y luego calcular la media de esas funciones por trozos. Como hemos visto, para representar esas cadenas mediante funciones necesitamos encontrar una parametrización común para las mismas de manera que sea posible alinearlas y

calcular su media. Hemos presentado dos alternativas: Una primera, a la que hemos llamado Método-X, utiliza los valores del eje de las x como parametrización común, y aunque *a priori* era la aproximación más natural, hemos visto que daba problemas ya que a veces para un mismo valor x teníamos más de un posible valor y y por tanto no podíamos transformar la cadena en una función. Por ello hemos presentado una segunda solución, que hemos llamado Método-L, ésta se basa en utilizar como parametrización común el porcentaje de longitud de las aristas que forman la cadena. Este último método es el que consideramos correcto. Todo el proceso del cálculo de la cadena media se ha incorporado al bien conocido algoritmo de programación dinámica que calcula el coste medio de la secuencia de edición entre cadenas.

La contribución del trabajo del cálculo de la forma media debe verse desde el punto de vista de su utilidad práctica. Primero, los algoritmos que se encuentran en la literatura para el cálculo de la cadena media asumen que el alfabeto de símbolos que pueden tener es finito, mientras que en el presente trabajo se considera que es un alfabeto infinito. En segundo lugar el coste computacional del algoritmo que calcula la cadena media crece de forma exponencial con el número de cadenas. En este trabajo se ha reducido el coste a polinomial, realizando una solución aproximada del mismo que calcula la cadena media de una manera incremental. Por otro lado la optimización del resultado depende también de la definición de las operaciones de edición, en función de los atributos geométricos, especialmente de la operación de fusión.

Para terminar debemos remarcar que el algoritmo del cálculo de la forma media ha sido evaluado experimentalmente, no sólo en los ejemplos presentados en este capítulo sino en el contexto presentado en el Capítulo 5. Aunque no podemos dar una evaluación cuantitativa completa dado que la distancia que utilizamos en la ecuación 2.9 no es una distancia real, sino una aproximación, si podemos dar una evaluación práctica del algoritmo utilizándolo en un proceso de segmentación de formas similares en el que el cálculo de la forma media de las mismas es un punto clave para la correcta segmentación de las mismas. Este proceso se lleva a cabo en un sistema de interpretación de documentos como el presentado en [34], y explicado detalladamente en el capítulo 5. En este entorno, una vez que una textura que consiste en una repetición regular de formas similares, se detecta, se calcula la forma media que representa de forma compacta el área texturada. En este contexto, se ha testeado el algoritmo en más de treinta documentos con diferentes tipos de texturas y formas, y los resultados se han considerado suficientemente buenos, de acuerdo con los resultados esperados *a priori*.

