



UNIVERSITAT DE VALÈNCIA  
Departament d'Informàtica i Electrònica

# CONTORNOS ACTIVOS: BASES TEÓRICAS, REPRESENTACIÓN Y MINIMIZACIÓN

**Tesina**

**Septiembre 1999**

Presentada por: *Juan Gutiérrez Aguado*  
Dirigida por: *Francesc J. Ferri Rabasa*



## Índice general

<i>Índice.</i> . . . . .	II
<i>Figuras.</i> . . . . .	IV
<i>Tablas.</i> . . . . .	V
<i>1.. Introducción</i> . . . . .	1
<i>2.. La ecuación del snake</i> . . . . .	5
2.1. Problemas inversos . . . . .	5
2.1.1. La derivación numérica como un problema inverso. . . . .	6
2.2. Problemas mal planteados. . . . .	7
2.3. Teoría de la regularización . . . . .	10
2.3.1. Ejemplo: derivación numérica unidimensional. . . . .	11
2.3.2. La ecuación del snake . . . . .	13
<i>3.. Representación del snake.</i> . . . . .	19
3.1. Puntos aislados. . . . .	19
3.2. Representación mediante B-Splines. . . . .	20
3.2.1. Aproximación mediante B-Splines . . . . .	20
3.2.2. Parametrización del snake usando B-splines . . . . .	29
<i>4.. Energía externa. Minimización de la energía</i> . . . . .	31
4.1. Energía externa . . . . .	31
4.2. Minimización de la energía . . . . .	33
4.2.1. Stifness matrix . . . . .	33

4.2.2. Método de Kass . . . . .	34
4.2.3. Solución por sobrerrelajación sucesiva. . . . .	34
4.2.4. Algoritmos voraces . . . . .	35
5.. <i>Experimentos y resultados.</i> . . . . .	39
5.1. Experimentos . . . . .	39
5.2. Resultados experimentales. . . . .	42
5.2.1. Resultados sobre las imágenes sintéticas. . . . .	42
5.2.2. Resultados sobre las imágenes reales. . . . .	50
5.2.3. Comentarios. . . . .	54
6.. <i>Conclusiones</i> . . . . .	57
A.. <i>Cálculo de Variaciones.</i> . . . . .	59
B.. <i>Discretización de la ecuación del snake.</i> . . . . .	63

## Índice de figuras

2.1. Rango y Núcleo . . . . .	9
2.2. Datos y datos con ruido . . . . .	13
2.3. Diferencias finitas y regularización. . . . .	14
2.4. Error en diferencias finitas y regularización . . . . .	15
3.1. Funciones base de grado cero. a) $B_{0,0}$ , b) $B_{1,0}$ , c) $B_{2,0}$ y d) $B_{3,0}$ . . . . .	22
3.2. Funciones base de grado uno. a) $B_{0,1}$ , b) $B_{1,1}$ , c) $B_{2,1}$ . . . . .	24
3.3. Funciones base de grado dos. a) $B_{0,2}$ y b) $B_{1,2}$ . . . . .	25
3.4. Función base $B_{0,3}(t)$ . . . . .	26
3.5. Partes de la función base $B_{0,3}(t)$ . . . . .	26
3.6. Derivada de la función $B_{0,3}$ . a) $B'_{0,3}$ y b) $B''_{0,3}$ . . . . .	27
3.7. Muestras de la función base $B_{0,3}(t)$ con $\delta = 0,2$ . . . . .	28
4.1. $- G_\sigma * \nabla I(x, y) $ y chamfer . . . . .	32
4.2. Ventana de búsqueda cuadrada. . . . .	37
4.3. Ventana de búsqueda lineal . . . . .	37
5.1. Imágenes sintéticas. a) Lab1, b) Lab2, c) Lab3 y d) Lab4 . . . . .	40
5.2. Imagen real: llave . . . . .	41
5.3. Imagen real: coche . . . . .	41
5.4. Snake para Lab1 con $- G_\sigma * \nabla I $ . . . . .	43
5.5. Snake para Lab1 con distancia chamfer . . . . .	44
5.6. Snake para Lab2 con $- G_\sigma * \nabla I $ . . . . .	45
5.7. Snake para Lab2 con distancia chamfer . . . . .	46
5.8. Snake para Lab3 con $- G_\sigma * \nabla I $ . . . . .	47

5.9. Snake para Lab3 con distancia chamfer . . . . .	48
5.10. Snake para Lab4 con $- G_\sigma * \nabla I $ . . . . .	49
5.11. Snake para Lab4 con distancia chamfer . . . . .	50
5.12. Snake para Llave con $- G_\sigma * \nabla I $ . . . . .	51
5.13. Snake para Llave con distancia chamfer . . . . .	52
5.14. Snake para Coche con $- G_\sigma * \nabla I $ . . . . .	53
5.15. Snake para Coche con distancia chamfer . . . . .	54

## Índice de cuadros

5.1. Tiempos para Lab1 con $- G_\sigma * \nabla I $ . . . . .	42
5.2. Tiempos para Lab1 con distancia chamfer. . . . .	43
5.3. Tiempos para Lab2 con $- G_\sigma * \nabla I $ . . . . .	44
5.4. Tiempos para Lab2 con distancia chamfer. . . . .	45
5.5. Tiempos para Lab3 con $- G_\sigma * \nabla I $ . . . . .	46
5.6. Tiempos para Lab3 con distancia chamfer. . . . .	47
5.7. Tiempos para Lab4 con $- G_\sigma * \nabla I $ . . . . .	48
5.8. Tiempos para Lab4 con distancia chamfer. . . . .	49
5.9. Tiempos para la llave con $- G_\sigma * \nabla I $ . . . . .	51
5.10. Tiempos para la llave con distancia chamfer. . . . .	52
5.11. Tiempos para el coche con $- G_\sigma * \nabla I $ . . . . .	53
5.12. Tiempos para el coche con distancia chamfer. . . . .	54



## 1. INTRODUCCIÓN

La visión comienza con la transformación de un flujo de fotones en un conjunto de valores de intensidad. El primer paso en el procesado visual de la información es la obtención de una descripción compacta de los valores de intensidad tal cual han sido obtenidos. Esta descripción debe ser **completa** en tanto en cuanto debe representar toda la información contenida en la imagen y además debe tener significado.

Los bordes físicos son una de las características más importantes de los objetos ya que se corresponden con contornos, con cambios en la orientación de superficies o cambio de las propiedades del material. Los bordes tridimensionales se transforman en la imagen en puntos críticos de la intensidad 2-D formada en el ojo o en la cámara. Estos puntos críticos son aquellos en los que se producen cambios en los valores de intensidad. Un borde por tanto debe poseer las dos propiedades siguientes: debe ser suave o suave a trozos y el gradiente de la imagen a lo largo del borde debe ser grande. La primera parte, por tanto, de la obtención de bordes requiere la evaluación de las derivadas de la intensidad. Puesto que la imagen es un conjunto discreto de valores de intensidad, la detección de bordes es un problema de derivación numérica. Este proceso se realiza habitualmente mediante la convolución de un determinado núcleo con la imagen siendo los más conocidos los de Sobel, Prewitt, ... El resultado de este primer paso es un conjunto de bordes (segmentos y curvas) que probablemente serán inconexos, por tanto el siguiente paso será la unión de ellos de tal forma que se agrupen formando los contornos de los objetos presentes en la escena.

Como se verá en uno de los capítulos, la derivación numérica puede ser vista como un problema inverso mal planteado en el sentido de Hadamard. Los problemas mal planteados se pueden convertir en problemas bien planteados mediante el uso de la teoría de la regularización añadiendo un término de estabilización de la solución (estabilizadores de Tikhonov).

Otra aproximación que conduce a la obtención de los contornos de los objetos es el uso de contornos activos [11] que se suele plantear como un problema de mecánica clásica (mediante potenciales) o como un problema de la teoría de la aproximación (mediante la teoría de la regularización).

Los contornos activos o *snakes* se definen como una serie de puntos y una ecuación variacional a minimizar partiendo de una iniciación externa al objeto del cual se quiere extraer el contorno. La minimización de la ecuación variacional se puede

abordar desde diversos puntos de vista (técnicas de análisis numérico -diferencias finitas, elementos finitos-, técnicas algorítmicas - programación dinámica, algoritmos voraces-, técnicas conexionistas -redes neuronales de Hopfield o técnicas probabilísticas -simulated annealing). En este trabajo nos centraremos en el estudio de la minimización mediante diferencias finitas y mediante algoritmos voraces.

Según han señalado algunos autores [6] los modelos usados para la detección de contornos deben poseer las siguientes características:

- Encontrar los contornos globales que mejor reproducen los contornos aparentes en la imagen.
- Capacidad de extraer formas suaves (contornos regulares).
- Respetar algunas singularidades: ángulos, esquinas, ...
- Robustez de los algoritmos: estabilidad y convergencia con un conjunto mínimo de parámetros.
- Adaptación eventual para encontrar varios contornos simultáneamente.
- Buena base teórica del método.

Como veremos a lo largo de este trabajo, de todos los puntos citados en el que más falla el modelo de contornos activos es en la capacidad de encontrar varios contornos simultáneamente, de todos modos esto tampoco debería ser mayor inconveniente ya que una vez detectados los objetos se podrían obtener los contornos secuencialmente.

Los objetivos son presentar formalmente la ecuación del snake como un caso particular de la teoría de la regularización [2] usando un determinado tipo de estabilizador de Tikhonov [16] desarrollando lo que en el artículo de Kass [11] se menciona pero sin presentar su desarrollo. Por otra parte se pretende mostrar cómo puede ajustarse la ecuación del snake a la representación que de la curva se escoja realizando una representación matricial para el caso de que la representación se realice mediante B-splines (polinomios cúbicos definidos a trozos usados para aproximar una serie de puntos). Finalmente se mostrarán dos posibilidades a la hora de escoger la energía externa (que es el término dentro de la ecuación del snake responsable de atraerlo hacia los bordes) y tres posibilidades de minimización de la ecuación. Con todo ello se presente dar una visión general de lo que es la teoría de contornos activos o snakes.

El presente trabajo se organiza del siguiente modo: en el capítulo segundo se presentan las bases de los problemas inversos y mal planteados, se mostrará que la derivación numérica puede considerarse como un problema inverso y mediante un ejemplo se verá que este planteamiento tiene muchas ventajas respecto a la formulación directa. Se verá un método para solucionar los problemas inversos basado en

estabilizadores de Tikhonov y de este modo se obtendrá la ecuación del snake. En el capítulo tercero se centrará la atención en la representación del snake y se verán dos posibilidades, como puntos aislados o mediante un B-spline y se obtendrán las ecuaciones que gobiernan el snake en ambos casos. En el capítulo cuarto se presentarán dos opciones a la hora de elegir la energía externa: una basada en la convolución de una máscara gaussiana con el gradiente de la imagen y otra basada en la transformación de distancia o distancia chamfer. Para finalizar este capítulo se presentarán tres modos de minimizar la energía: mediante la formulación del artículo original de Kass, mediante sobrerrelajación sucesiva y mediante algoritmos voraces. En el capítulo quinto se presentan los experimentos a realizar y los resultados obtenidos y para finalizar en el capítulo sexto se presentan las conclusiones.



## 2. LA ECUACIÓN DEL SNAKE

En este capítulo se definirán los problemas inversos y mal planteados en el sentido de Hadamard. Se mostrará como puede formularse la derivación numérica como un problema inverso mediante una ecuación integral de Fredholm de primera especie. Este tipo de ecuaciones son mal planteadas y se verá cómo haciendo uso de estabilizadores de Tikhonov se llega a la ecuación del *snake*. Para finalizar se muestra un ejemplo de regularización de la derivación numérica.

### 2.1. Problemas inversos

Sea  $g$  el resultado de un determinado experimento que puede depender de una o varias variables. El propósito de este experimento es la estimación, mediante el adecuado procesamiento de la **función de datos**  $g$ , de alguna característica física o geométrica de la muestra que, en general, puede ser vista como una función  $f$  de argumento real o vectorial. Llamaremos a  $f$  **la función objeto**. En el caso concreto que nos ocupa  $g$  será la imagen y  $f$  los bordes que queremos determinar.

El **problema directo** consiste en la determinación de la aplicación desde el conjunto de todos los posibles objetos en el conjunto de todos los posibles datos.

Si  $f \in F$  y  $g \in G$ , donde habitualmente  $F$  y  $G$  son dos espacios de Hilbert, el problema directo es obtener  $g$  dado un operador  $B$  y dado  $f$ :

$$B : F \rightarrow G \text{ tal que } Bf = g \quad (2.1)$$

El **problema inverso** consiste en la determinación del objeto  $f$  a partir de los datos  $g$ . Corresponde a la inversión de la aplicación directa, que habitualmente es no lineal, y por tanto el problema inverso implica la solución de una ecuación funcional no lineal. En la mayoría de los casos es posible y razonable, sobre la base de aproximaciones físicas apropiadas, linealizar el problema. Por tanto, el resultado es una ecuación funcional de primera clase. El problema inverso se puede formular del siguiente modo: dados  $A$  y  $g$  encontrar  $f$  tal que:

$$Af = g$$

Para finalizar esta sección se darán algunos ejemplos de problemas inversos:

Por ejemplo cuando se intenta ajustar un modelo a unos datos dados, el planteamiento es el siguiente: de acuerdo con algún modelo teórico, el valor de una cantidad  $y$  depende de otra cantidad  $x$  según la ecuación  $y = a + bx + cx^2 + dx^3$ . Dado un conjunto de puntos  $(x_i, y_i)$  (que son los datos) ¿Cómo determinar los valores  $a, b, c$  y  $d$ ? En este caso el objeto son los números  $a, b, c$  y  $d$ .

Otro ejemplo de problema inverso es el de deconvolución. Dada una fotografía borrosa, o el resultado del paso de una señal por un medio que actúa como un filtro, ¿cómo se puede reconstruir una versión clara de la imagen borrosa, o de la señal original antes de ser filtrada?

### 2.1.1. La derivación numérica como un problema inverso.

Supóngase una imagen  $g$  de una escena, o en otras palabras, los datos. Se desea obtener la derivada  $f$  de estos datos. Según la definición previa este es un problema inverso que se formula del siguiente modo:

$$\int_{\Omega} K(x, s)f(s)ds = g(x) \quad (2.2)$$

La ecuación 2.2 es una ecuación integral de Fredholm de primera especie y para el caso de la derivación el núcleo es la función salto unidad de Heaviside:

$$K(x, s) = U(x - s)$$

Para simplificar, aunque sin perder generalidad, supongamos  $\Omega = [a, b]$  y la ecuación 2.2 puede escribirse como:

$$\int_a^b U(x - s)f(s)ds = \int_a^x f(s)ds = g(x)$$

Si  $F(x)$  es la primitiva de  $f(x)$ , por el teorema fundamental del cálculo se tiene:

$$F(x) - F(a) = g(x)$$

derivando respecto de  $x$  se obtiene:

$$F'(x) = g'(x) \text{ o lo que es lo mismo } f(x) = g'(x)$$

es decir, se ha mostrado la equivalencia entre la ecuación integral y la ecuación diferencial.

En la última sección se verán las ventajas de la ecuación integral frente a la ecuación diferencial, aunque de un modo intuitivo se puede decir que a pesar de que el resultado de la integración da como resultado funciones más complejas estas poseen mejores propiedades (por ejemplo en cuanto a suavidad) que el resultado de la derivación.

## 2.2. Problemas mal planteados.

Supondremos en lo sucesivo que  $X$  es el espacio de objetos e  $Y$  es el espacio de datos, siendo ambos espacios de Hilbert y que  $A : X \rightarrow Y$  es un operador lineal y continuo.

Cualquiera que sea el operador  $A$ , el problema inverso

$$Af = g$$

para  $g$  dado, es un problema bien planteado en el sentido de Hadamard si:

1. Existe una solución para todo dato  $g$ .
2. La solución es única en el espacio imagen  $Y$ .
3. La aplicación inversa  $Y \rightarrow X$  es continua.

Si no se cumple alguna de estas propiedades se dice que el problema está **mal planteado**.

El concepto de problema mal planteado lo introdujo Hadamard en 1923 y en sus orígenes se pensaba este tipo de problemas no tenían sentido físico, prevaleciendo la opinión de que sólo tenían interés en las aplicaciones los problemas correctamente planteados, suposición que está muy lejos de la realidad, ya que, por ejemplo pueden ser inaccesibles los datos sobre alguna parte de la frontera (en el caso de una ecuación diferencial con condiciones de contorno) o, sencillamente, no se disponen de los datos necesarios, ni con la precisión suficiente, para resolver el problema. Es costumbre en la práctica o bien ignorar la cuestión de la existencia, o bien alterar el concepto de solución o variar el modelo matemático. En muchas situaciones prácticas se puede definir la "solución" como aquella función de alguna clase conveniente que mejor aproxima los datos o, alternativamente, emplear algún tipo de técnica de regularización que genere "pseudosoluciones" del problema en cuestión.

En esta sección comentaremos las definiciones básicas y bajo qué circunstancias un problema puede estar mal planteado.

Daremos algunas definiciones básicas sobre análisis funcional para estudiar cada uno de estos puntos que hacen que un problema esté mal planteado.

Definición 1: El **rango** de  $A$  es el conjunto de elementos de  $Y$  que son imagen de miembros de  $X$ , en otras palabras:

$$R(A) = \{g : g \in Y, Af = g \text{ para algún } f \in X\}$$

Si el rango de  $A$  es todo  $Y$  se dice que el operador es suprayectivo.

Definición 2: El **núcleo** de  $A$  es el conjunto de elementos del dominio de  $A$  cuya imagen es cero:

$$N(A) = \{f \in X : Af = 0\}$$

Estos dos conceptos están representados en la figura 2.1

Definición 3: Se define el **complemento ortogonal** de un subespacio cerrado  $V$  de un espacio de Hilbert  $X$  respecto al producto interno de  $X$  y que denotaremos por  $(,)_X$  como

$$V^\perp = \{u \in X : (u, v)_X = 0 \quad \forall v \in V\}$$

Teorema 1: Sea  $V$  un subespacio cerrado de un espacio de Hilbert  $X$ . Entonces todo  $u \in X$  puede escribirse en la forma:

$$u = v + w \quad u \in X, v \in V, w \in V^\perp$$

esto es  $X = V \oplus V^\perp$ <sup>1</sup>

Se dice que el operador  $A$  es **unívoco** si cada elemento de  $X$  solo tiene una imagen en  $Y$ . Si estamos en estas condiciones se puede definir un operador  $A^{-1}$  llamado inversa de  $A$  que lleva de  $Y$  a  $X$ :

$$A^{-1} : R(A) \rightarrow X \quad A^{-1}(Af) = f$$

Con estas definiciones podemos comentar cada uno de las condiciones que debe cumplir un problema bien planteado.

<sup>1</sup> Si  $V$  y  $W$  son subespacios de un espacio lineal  $X$  entonces  $X$  es la suma directa de  $V$  y  $W$  si  $U = V + W$  y  $V \cap W = 0$  y se representa mediante  $V \oplus W$

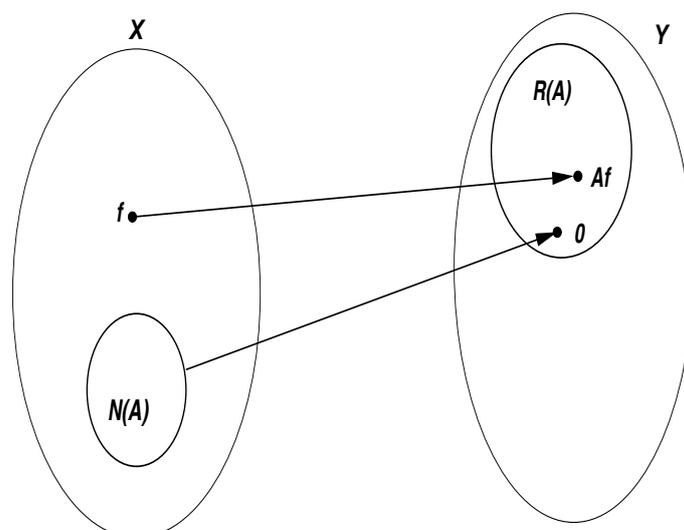


Fig. 2.1: Rango y Núcleo

### 1. Existencia.

Dada la ecuación  $Af = g$  con  $A : X \rightarrow Y$  y  $X, Y$  espacios de Hilbert, debemos determinar el conjunto de funciones  $g \in Y$  para las que la ecuación  $Af = g$  admite solución. Esto es, debemos identificar  $R(A)$ . Esto nos permitirá resolver el problema de la existencia de la solución. Si encontramos que  $R(A)$  no es todo el espacio  $Y$  entonces habrá funciones  $g \in Y$  que no están en  $R(A)$  para las cuales no existe solución.

### 2. Unicidad.

La segunda tarea es obtener bajo qué circunstancias la solución es única, en otras palabras, queremos conocer bajo qué condiciones  $A$  es biyectivo. Si  $N(A) \neq 0$  no tendremos una única solución ya que si  $u_0$  es una solución también lo será  $u_0 + u \quad \forall u \in N(A)$  ya que de la linealidad de  $A$  se desprende  $A(u_0 + u) = A(u_0) + A(u) = A(u_0) = g$ . Por tanto deberemos excluir los elementos de  $N(A)$  del dominio de  $A$  si queremos asegurar unicidad.

Puesto que hemos supuesto que los espacios  $X$  e  $Y$  son espacios de Hilbert tendremos definido un producto interno que en el caso del espacio  $X$  representaremos mediante  $(\cdot, \cdot)_X$ .

Puede mostrarse que  $N(A)$  es de dimensión finita y por tanto completo y cerrado, por lo que haciendo uso del teorema 1 tenemos que  $X = N(A) \oplus N(A)^\perp$  en otras palabras, todo  $f \in X$  es de la forma  $f = u + w$  con  $u \in N(A)$  y  $w \in N(A)^\perp$  y  $N(A) \cap N(A)^\perp = 0$ . Puesto que  $N(A)$  y  $N(A)^\perp$  son disjuntos,

excepto para el elemento cero, se puede restringir el dominio de  $A$  a  $N(A)^\perp$  para asegurar la unicidad.

### 3. Continuidad de la inversa.

Las condiciones 1 y 2 de Hadamard implican que el operador  $A$  tiene inversa  $A^{-1}$  y que  $D(A^{-1}) = Y$ . Si suponemos que  $A$  es lineal y continuo se puede deducir que  $A^{-1}$  es continuo, en el caso lineal la continuidad de la inversa es una consecuencia de 1 y 2. En otros casos habría que hacer uso de otros teoremas sobre operadores que se pueden encontrar por ejemplo en [15] [pag. 208-246].

Cuando alguna de estas condiciones no se cumple el problema  $Af = g$  está mal planteado. Es evidente por todo lo dicho que el mal planteamiento es una propiedad del triplete  $\{A, X, Y\}$ . El problema es mal planteado por que, por ejemplo el espacio  $Y$  es muy amplio (por tanto  $R(A) \neq Y$ ) y fallaría la existencia. En tal caso se puede intentar transformar el problema modificando el espacio  $Y$  y su topología<sup>2</sup>. En la mayoría de problemas sin embargo, este es un ejercicio matemático inútil ya que la elección apropiada del espacio de datos y de su topología no está dictada solamente por consideraciones matemáticas.

Todos estos conceptos han sido utilizados en la visión por computador y se encuentran aplicaciones en la detección de bordes [17], flujo óptico [12] y [3], reconstrucción de superficies [3] y [16], etc.

Para una aplicación de todos estos conceptos a ecuaciones diferenciales y problemas variacionales con condiciones de contornos un libro excelente es [13] [pag. 202-214 y 221-229]. Una descripción matemática rigurosa de los problemas inversos y mal planteados se puede encontrar en [2].

Con toda esta discusión se ha pretendido mostrar que haciendo uso del analisis funcional es relativamente sencillo caracterizar cuando un problema estará bien o mal planteado.

### 2.3. Teoría de la regularización

Como se ha comentado los problemas inversos tienden a ser problemas mal planteados por lo que la existencia, unicidad y estabilidad de la solución no puede ser garantizada en ausencia de restricciones. Una de las restricciones posibles es la de suavidad de la solución. Esta restricción ha sido utilizada en muchos problemas de visión aunque sin una justificación teórica. Sin embargo, una base formal para restricciones similares se puede encontrar en ciertas aproximaciones a la solución matemática de problemas inversos, particularmente la teoría de la regularización presentada por

<sup>2</sup> La topología de un espacio está determinada por la norma definida en el espacio

Tikhonov. La regularización de Tikhonov emplea *funcionales de estabilización* para restringir las soluciones admisibles a espacios de funciones “suaves”. Bajo condiciones no excesivamente restrictivas, los principios variacionales resultantes pueden resultar bien planteados en estos espacios y por tanto se puede obtener una solución.

Tikhonov propuso un estabilizador para la regularización que coincide con la norma de Sobolev  $p$ -ésima y ponderada:

$$\|v\|_p^2 = \sum_{|\alpha| \leq p} \|w_p D^\alpha v\|_{L^2}^2$$

donde  $w_p$  son funciones continuas y no negativas. Para el caso multidimensional se define

$$D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \dots \partial x_n^{\alpha_n}}$$

y  $\alpha$  es el conjunto de todas la  $n$ -tuplas ordenadas de enteros no negativos, es decir,  $\alpha = (\alpha_1, \dots, \alpha_n)$  y  $|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_n$ . Esta expresión para el caso unidimensional toma la siguiente forma:

$$\|v\|_p^2 = \sum_{\alpha=0}^p \int_{\mathcal{R}} w_\alpha(x) \left( \frac{d^\alpha v(x)}{dx^\alpha} \right)^2 dx$$

La teoría abstracta de aproximación óptima está bien desarrollada y está próxima a los principios variacionales que involucran la minimización con restricciones de normas en espacios de funciones de Hilbert.

A continuación se muestra un ejemplo de regularización unidimensional: la derivación numérica.

### 2.3.1. Ejemplo de regularización: derivación numérica unidimensional.

Supongamos dadas las muestras de una función  $g$  en un determinado intervalo  $[x_1, x_m]$  con un espaciado uniforme  $\delta$ . Si el objetivo es obtener la derivada en los puntos  $x_1, x_2, \dots, x_m$  y siempre que el espaciado sea pequeño, se puede pensar en obtener la derivada mediante las diferencias finitas:

$$f = Dg(x_i) = \frac{g(x_i) - g(x_{i+1})}{\delta}$$

Si las muestras de la función están libres de ruido, esta aproximación da buenos resultados siempre que se cumplan las condiciones descritas.

Supongamos ahora que las muestras tienen ruido y que además deseamos obtener la derivada en el intervalo  $[s_1, s_n]$  con un espaciado de puntos  $\epsilon < \delta$ . En este caso el esquema de diferencias finitas no es válido.

Enfoquemos la derivación como un problema inverso. Recuérdese que la función de datos y la función objeto están relacionadas mediante la ecuación:

$$Af = g$$

donde  $Af = \int_{s_1}^x f(s)ds = g(x)$ .

Esta ecuación se puede discretizar usando algún esquema de integración numérica. Por tanto transformamos  $A$  en una matriz  $\mathbb{R}^{m \times n}$  cuya inversa no se puede calcular. Si pensamos en obtener  $f$  como la solución al problema de mínimos cuadrados:

$$\|Af - g\|^2$$

llegaríamos a la siguiente ecuación:

$$f = (A^T A)^{-1} A^T g$$

donde la matriz  $(A^T A)^{-1} A^T$ , que se conoce como pseudoinversa de Moore-Penrose de  $A$ , no puede calcularse, por ser  $(A^T A)^{-1}$  singular.

La solución a este problema la proporciona la regularización.

Supongamos tal y como se comentó en la sección anterior que transformamos nuestro problema en este otro: encontrar  $f$  que minimiza la siguiente ecuación:

$$\|Af - g\|^2 + \lambda^2 \|Lf\|^2$$

donde  $L$  es un operador diferencial discreto, o una combinación lineal de operadores diferenciales discretos de diversos órdenes.

Se puede obtener una solución derivando respecto a  $f_k$  e igualando a cero:

$$\frac{\partial}{\partial f_k} \{ \lambda^2 f^T L^T L f + (g - Af)^T (g - Af) \} = 0$$

para  $k = 1, 2, \dots, n$ . Esto conduce al sistema de ecuaciones :

$$2\lambda^2 L^T L f - 2A^T (g - Af) = 0$$

o despejando  $f$ :

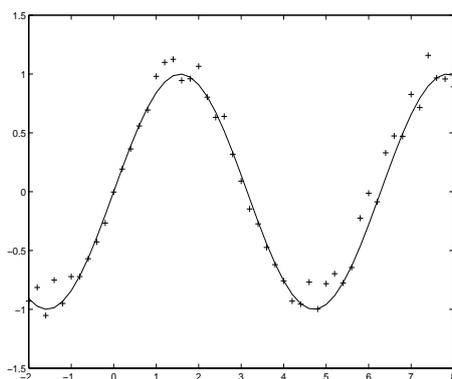


Fig. 2.2: Datos y datos con ruido

$$f = (\lambda^2 L^T L + A^T A)^{-1} A^T g$$

Con esto se obtiene una familia de soluciones en función de  $\lambda$  de tal forma que si  $\lambda = 0$  este sistema se reduce a la solución por mínimos cuadrados y para valores distintos de cero el término adicional  $\lambda^2 L^T L$  altera los autovalores y autovectores de la matriz  $A^T A$ . Siempre que  $\lambda^2 L^T L + A^T A$  sea no singular habrá una única solución.

Supongamos muestras de la función  $g = \sin(x)$  en el intervalo  $\Omega = [-2, 8]$  con un espaciado  $\delta = 0,2$ . Deseamos obtener la derivada en el intervalo  $\Upsilon = [-2, 8]$  con un espaciado  $\epsilon = 0,075$ . Supongamos además que las muestras tienen ruido. En la figura 2.2 se muestra la función  $\sin(x)$  y la función  $\sin(x) + \eta(x)$ , siendo  $\eta(x)$  el ruido.

Como se puede ver en la figura 2.3 el esquema de diferencias finitas es muy sensible al error en los datos, mientras que usando la regularización (siempre que el valor de  $\lambda$  sea el adecuado) se obtiene el mismo resultado que con los datos sin ruido.

El valor que mejor aproxima la derivada es el de  $\lambda = 0,1$ , para valores más bajos de  $\lambda$  el método es todavía sensible al ruido, mientras que para valores altos se atiende más a la suavidad que a los datos.

En la figura 2.4 se muestra el error de cada uno de los dos métodos.

### 2.3.2. La ecuación del snake

Para finalizar este capítulo se llegará a la ecuación del *snake* partiendo de lo visto sobre regularización.

Sea  $X$  un espacio lineal de funciones suaves admisibles. Sea  $\mathcal{S}(v) : X \rightarrow \mathbb{R}$  un funcional definido sobre  $X$  que mide la suavidad de una función admisible  $v \in X$ .

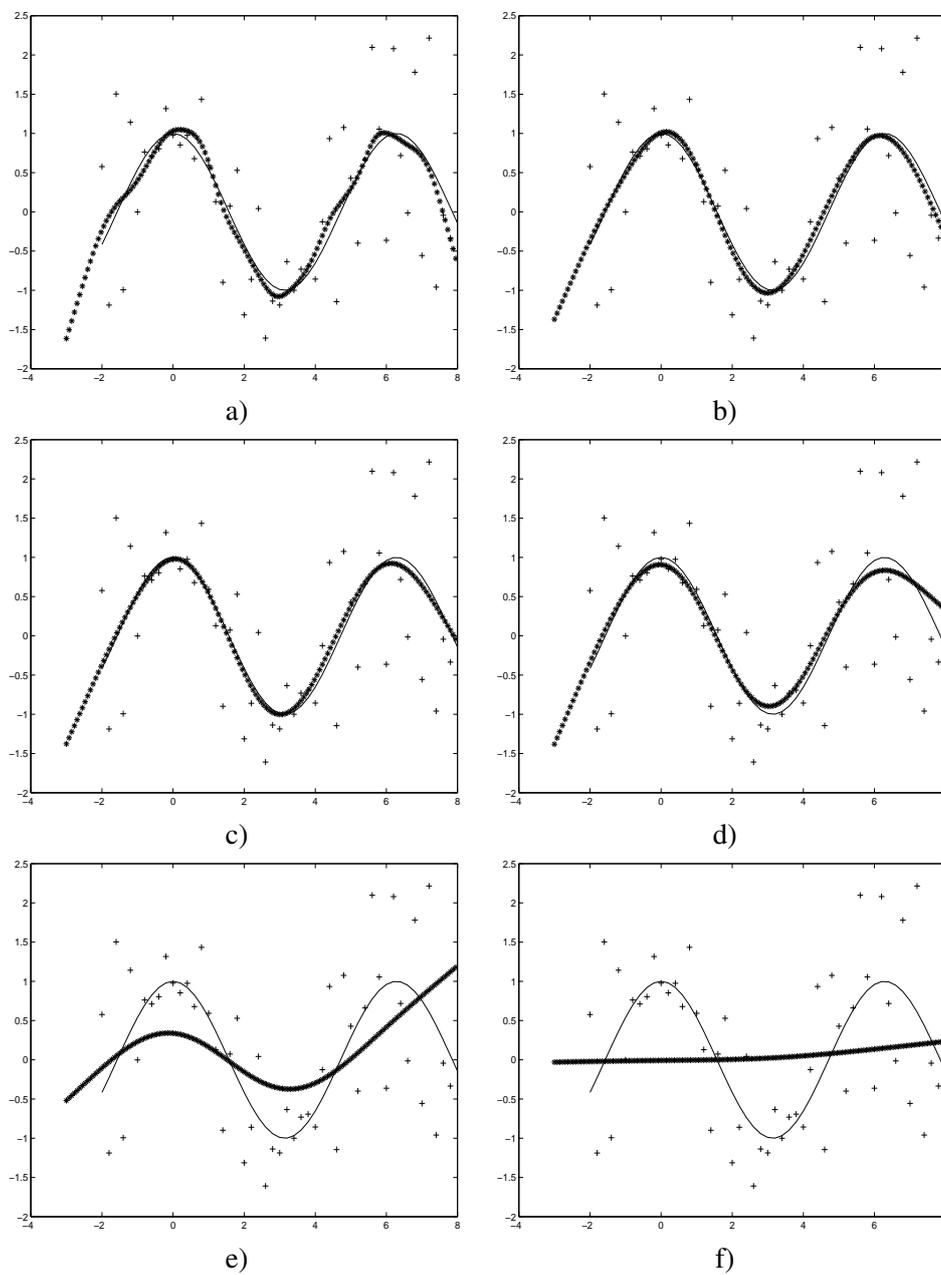


Fig. 2.3: Diferencias finitas y regularización. a)  $\lambda = 0,01$ , b)  $\lambda = 0,05$ , c)  $\lambda = 0,1$ , d)  $\lambda = 0,2$ , e)  $\lambda = 1$  y f)  $\lambda = 10$

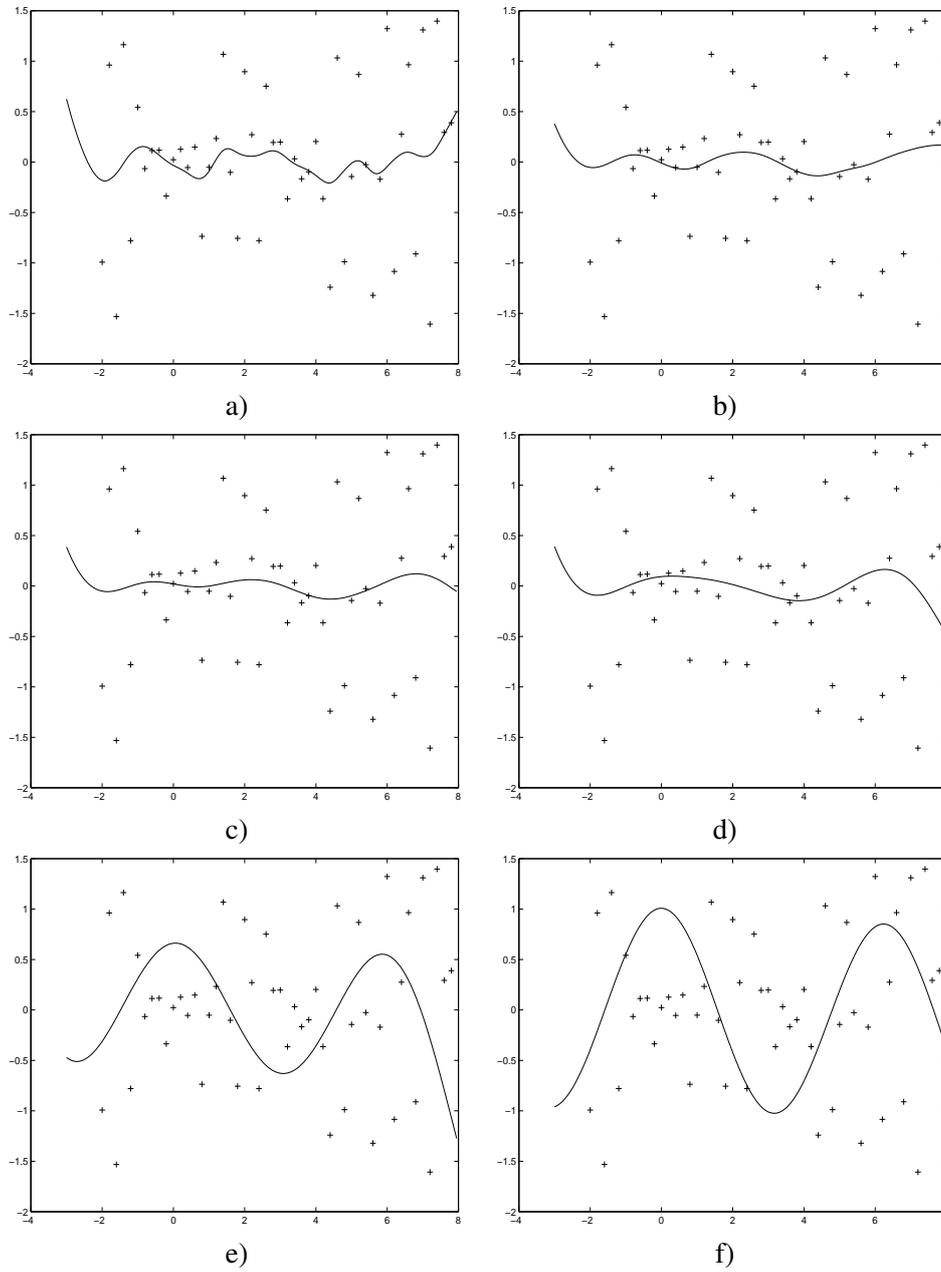


Fig. 2.4: Error en diferencias finitas y regularización. a)  $\lambda = 0,01$ , b)  $\lambda = 0,05$ , c)  $\lambda = 0,1$ , d)  $\lambda = 0,2$ , e)  $\lambda = 1$  y f)  $\lambda = 10$

Además sea  $\mathcal{P}(v) : X \rightarrow \mathbb{R}$  un funcional en  $X$  que mide la discrepancia entre la función y los datos dados.

Considérese el siguiente principio variacional:

Encontrar  $u \in X$  tal que

$$\mathcal{E}(u) = \underset{v \in X}{\text{mínimo}} \mathcal{E}(v)$$

donde el funcional  $\mathcal{E}$  viene dado por

$$\mathcal{E}(v) = \mathcal{S}(v) + \mathcal{P}(v)$$

Esto define un problema de aproximación óptima: encontrar la función admisible suave en  $X$  que es además compatible con los datos. La condición necesaria (pero en general no suficiente) que satisface la función  $u(x)$  que minimiza el funcional está dada por la anulación de la primera variación  $\delta$ , que expresan las conocidas ecuaciones de Euler-Lagrange  $\delta\mathcal{E}(u) = \delta\mathcal{S}(u) + \delta\mathcal{P}(u) = 0$ .

Si escogemos como funcional de penalización  $\mathcal{P}$  la integral del gradiente (con signo negativo) a lo largo de una curva, este será mínimo cuando el gradiente sea máximo, es decir:

$$\mathcal{P}(v) = - \int_{s_a}^{s_b} |\nabla I(v(s))| ds$$

y como funcional de suavidad la norma de Sobolev,

$$\mathcal{S}(v) = \int_{s_a}^{s_b} \left( \alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{d^2v}{ds^2} \right|^2 \right) ds$$

obtenemos la ecuación del *snake* tal y como apareció en el artículo original de Kass.

Para finalizar resumiremos todo lo dicho en esta sección:

- Se ha mostrado la naturaleza **inversa** de la derivación numérica.
- Como casi todos los problemas inversos la derivación numérica es un **problema mal planteado** en el sentido de Hadamard.
- Mediante el uso del **análisis funcional** se puede ver cuando un problema está mal planteado en función del operador y de su inversa.
- Se pueden usar técnicas de **regularización** para restringir el espacio de búsqueda al de funciones con una cierta “suavidad” al mismo tiempo que se ajustan a los datos.

- 
- Finalmente hemos llegado a la ecuación del *snake* sin más que adoptar un determinado estabilizador de Tikhonov.

En el siguiente capítulo se verán dos métodos de representar el *snake* y se ajustará la ecuación a la representación escogida.



### 3. REPRESENTACIÓN DEL SNAKE.

En el capítulo anterior se ha obtenido la ecuación del snake pero no se ha comentado nada sobre cómo se representará la curva que describe el contorno  $v(s)$  y cuya energía habrá que minimizar. En este capítulo hablaremos de dos posibles representaciones:

- Como un conjunto de puntos aislados en los que se define la citada energía.
- Como un conjunto de puntos que representan los puntos de control de algún tipo de curva paramétrica. En nuestro caso se han escogido los B-splines cúbicos que aproximan los puntos de control. La energía en este caso se define sobre la curva definida por el B-spline.

El contorno es una aplicación:

$$\begin{aligned}\Omega = [a, b] &\longrightarrow \mathbb{R}^2 \\ s &\longrightarrow v(s) = (x(s), y(s))\end{aligned}$$

Sobre el que se define el funcional  $E$  a minimizar:

$$v(s) \longrightarrow E_{snake}(v(s)) = \int_{\Omega} E_{int}(v) + E_{ext}(v) ds \quad (3.1)$$

El objetivo será adecuar la ecuación de la energía de la forma más sencilla posible a la representación elegida.

Comenzaremos con la primera de las representaciones comentadas.

#### 3.1. Puntos aislados.

Esta es la forma más sencilla de representar el contorno, y es la que apareció en el artículo original [11]. Supongamos que mediante algún proceso se ha obtenido

un contorno inicial que engloba al que se desea obtener. Este contorno estará representado mediante una serie de puntos  $v_i = (x_i, y_i)$   $i = 1, \dots, N$ . Por tanto hay que discretizar la ecuación 3.1.

Si se usan diferencias finitas para aproximar la primera y segunda derivada se obtiene la ecuación:

$$E_{snake} = \sum_{i=1}^N E_{int}(i) + E_{ext}(i) \quad (3.2)$$

donde

$$E_{int}(i) = \alpha_i \frac{|v_i - v_{i-1}|^2}{2h^2} + \beta_i \frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{2h^4}$$

y por ser un contorno cerrado se define  $v(0) = v(n)$ ,  $v(n+1) = v(1)$ .

El problema que tiene esta representación es que si se disponen de pocos puntos de control,  $N \ll \text{Longitud}(v)$ , el resultado puede ser impreciso. En la siguiente sección veremos como obtener una descripción del contorno usando B-Splines.

### 3.2. Representación mediante B-Splines.

En esta sección comenzaremos describiendo qué es un B-spline para posteriormente usarlo en la parametrización del snake.

#### 3.2.1. Aproximación mediante B-Splines

Dados  $n$  puntos  $(x_0, y_0) \dots (x_{n-1}, y_{n-1})$  (que se denominan **puntos de control**) se desea obtener una curva que se ajuste con la forma definida por esos puntos. Si se requiere que la curva pase a través de todos los puntos, se trata de un problema de *interpolación*. Si sólo se necesita que la curva esté cercana a estos puntos se trata de un problema de *aproximación*. Nos centraremos en el problema de la aproximación ya que si los datos iniciales tienen ruido la aproximación dará una curva suave. Para ello se obtendrá la curva como una combinación lineal de funciones base pesadas con los puntos que se desea aproximar:

$$Q(t) = (x(t), y(t)) = \left( \sum_{i=0}^{n-1} x_i B_{i,3}(t), \sum_{i=0}^{n-1} y_i B_{i,3}(t) \right) \quad (3.3)$$

donde las funciones  $B_{i,k}(t)$  son las funciones base con  $i$  indicando el punto de control y  $k$  el grado. Tal y como se observa en la ecuación 3.3 usaremos un polinomio cúbico.

Para el conjunto de nudos  $u_0, u_1, \dots$  se definen las funciones base  $B_{i,k}(t)$  de forma recurrente según la siguiente regla [7]:

$$B_{i,0}(t) = \begin{cases} 1, & \text{si } u_i \leq t \leq u_{i+1} \\ 0, & \text{en otro caso.} \end{cases}$$

$$B_{i,k}(t) = \frac{t - u_i}{u_{i+k} - u_i} B_{i,k-1}(t) + \frac{u_{i+k+1} - t}{u_{i+k+1} - u_{i+1}} B_{i+1,k-1}(t)$$

En el presente trabajo nos centraremos en contornos cerrados y con un conjunto de nudos uniformemente espaciados (lo cual permitirá obtener una función cúbica definida a trozos).

Las propiedades que presentan los *B-splines* son aptas para la descripción de contornos ya que:

- Son funciones  $C^2$ .
- Presentan un *control local*. Es decir, las funciones base son de soporte compacto por lo que cada segmento de la curva depende solamente de un subconjunto de los puntos de control.
- Son funciones de aproximación, y por tanto el contorno que definen es suave.

El objetivo ahora es construir una función definida a trozos en el dominio de definición del parámetro de la curva  $t$  teniendo en cuenta un conjunto de nudos uniformemente espaciados.

#### *Funciones base de grado 0.*

Las funciones  $B_{0,0}(t)$ ,  $B_{1,0}(t)$ , y  $B_{2,0}(t)$  y  $B_{0,3}(t)$  a partir de la definición recursiva, son

$$B_{0,0}(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq 1 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.4)$$

$$B_{1,0}(t) = \begin{cases} 1 & \text{si } 1 \leq t \leq 2 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.5)$$

$$B_{2,0}(t) = \begin{cases} 1 & \text{si } 2 \leq t \leq 3 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.6)$$

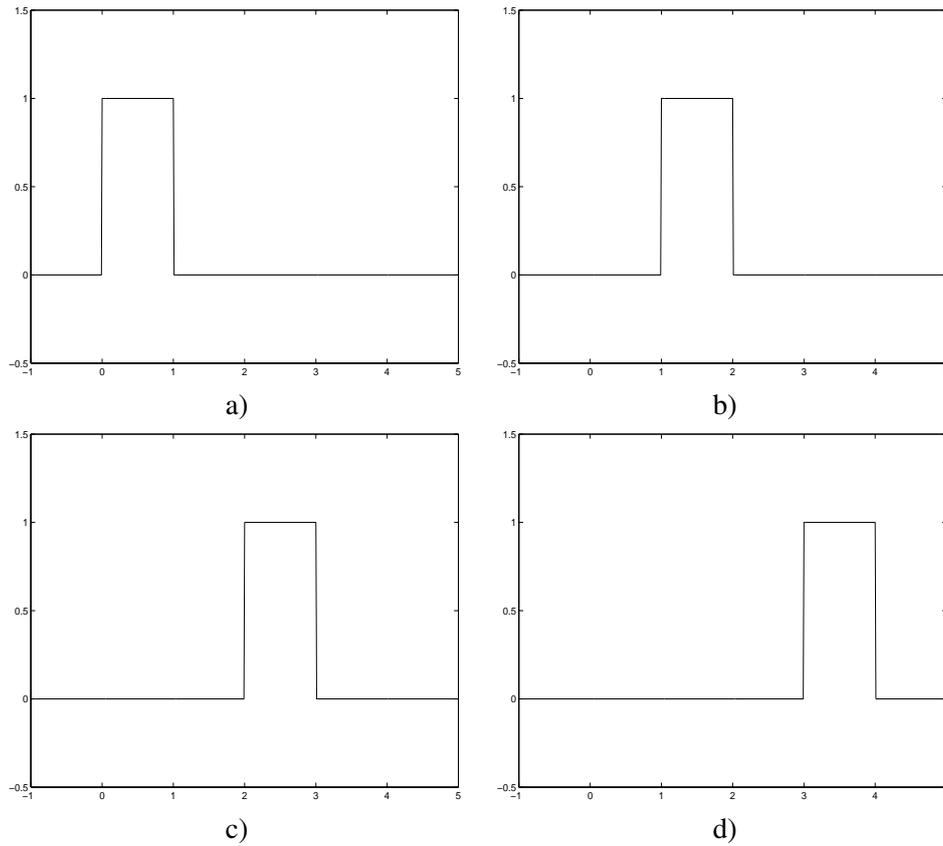


Fig. 3.1: Funciones base de grado cero. a)  $B_{0,0}$ , b)  $B_{1,0}$ , c)  $B_{2,0}$  y d)  $B_{3,0}$

$$B_{3,0}(t) = \begin{cases} 1 & \text{si } 3 \leq t \leq 4 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.7)$$

Estas funciones se muestran en la figura 3.1.

#### Funciones base de grado 1.

A partir de las anteriores podemos obtener la ecuación de las funciones  $B_{0,1}(t)$ ,  $B_{1,1}(t)$  y  $B_{2,1}(t)$ :

$$B_{0,1}(t) = \frac{t - u[0]}{u[1] - u[0]} B_{0,0}(t) + \frac{u[2] - t}{u[2] - u[1]} B_{1,0}(t)$$

Usando las ecuaciones 3.4 y 3.5 se obtiene

$$B_{0,1}(t) = \begin{cases} t & \text{si } 0 \leq t < 1 \\ 2 - t & \text{si } 1 \leq t < 2 \\ 0 & \text{en otro caso} \end{cases} \quad (3.8)$$

Del mismo modo se calcula  $B_{1,1}(t)$ :

$$B_{1,1} = (t) \frac{t - u[1]}{u[2] - u[1]} B_{1,0}(t) + \frac{u[3] - t}{u[3] - u[1]} B_{2,0}(t)$$

Usando las ecuaciones 3.5 y 3.6 se obtiene

$$B_{1,1}(t) = \begin{cases} t - 1 & \text{si } 1 \leq t < 2 \\ 3 - t & \text{si } 2 \leq t < 3 \\ 0 & \text{en otro caso} \end{cases} \quad (3.9)$$

Finalmente se obtiene la función  $B_{2,1}(t)$

$$B_{2,1}(t) = \frac{t - u[2]}{u[3] - u[2]} B_{2,0}(t) + \frac{u[4] - t}{u[4] - u[3]} B_{3,0}(t)$$

Usando las ecuaciones 3.6 y 3.7 se obtiene

$$B_{2,1}(t) = \begin{cases} t - 2 & \text{si } 2 \leq t < 3 \\ 4 - t & \text{si } 3 \leq t < 4 \\ 0 & \text{en otro caso} \end{cases} \quad (3.10)$$

Se puede observar que fijado el grado de la función base la siguiente se calcula del siguiente modo

$$B_{i+m,k}(t) = B_{i,k}(t - m) \quad (3.11)$$

es decir, es la misma función pero desplazada.

Las gráficas de estas funciones se muestran en la figura 3.2.

### Funciones base de grado 2

Ahora se construirán a partir de las anteriores  $B_{0,2}(t)$  y  $B_{1,2}(t)$

$$B_{0,2}(t) = \frac{t - u[0]}{u[2] - u[0]} B_{0,1}(t) + \frac{u[3] - t}{u[3] - u[1]} B_{1,1}(t)$$

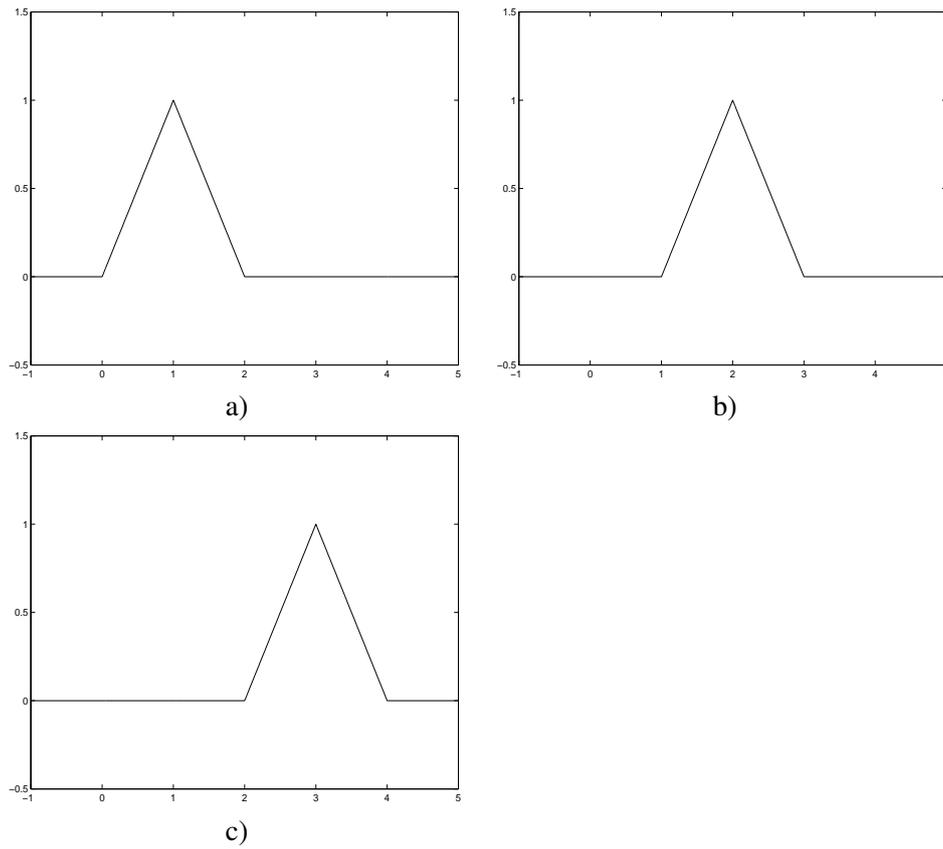


Fig. 3.2: Funciones base de grado uno. a)  $B_{0,1}$ , b)  $B_{1,1}$ , c)  $B_{2,1}$

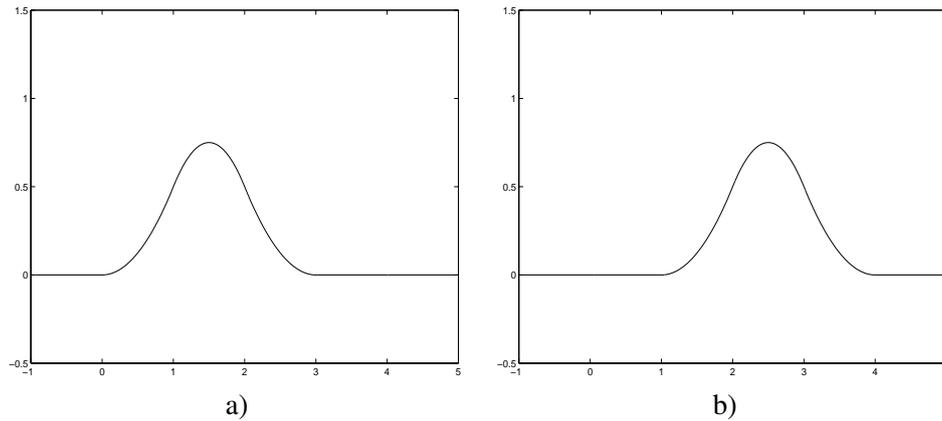


Fig. 3.3: Funciones base de grado dos. a)  $B_{0,2}$  y b)  $B_{1,2}$

Usando las ecuaciones 3.8 y 3.9 se llega a:

$$B_{0,2}(t) = \begin{cases} \frac{t^2}{2} & \text{si } 0 \leq t < 1 \\ \frac{-2t^2+6t-3}{2} & \text{si } 1 \leq t < 2 \\ \frac{(3-t)^2}{2} & \text{si } 2 \leq t < 3 \\ 0 & \text{en otro caso} \end{cases} \quad (3.12)$$

Si usamos la propiedad descrita en la ecuación 3.11 se obtendrá  $B_{1,2}(t)$  a partir de  $B_{0,2}(t)$  desplazando ésta una unidad:

$$B_{1,2}(t) = \begin{cases} \frac{(t-1)^2}{2} & \text{si } 1 \leq t < 2 \\ \frac{-2t^2+10t-11}{2} & \text{si } 2 \leq t < 3 \\ \frac{(4-t)^2}{2} & \text{si } 3 \leq t < 4 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.13)$$

Las gráficas de estas ecuaciones se muestran en la figura 3.3.

### Funciones base de grado 3

Finalmente a partir de las dos anteriores obtenemos la función cúbica  $B_{0,3}(t)$

$$B_{0,3}(t) = \frac{t - u[0]}{u[3] - u[0]} B_{0,2}(t) + \frac{u[4] - t}{u[4] - u[1]} B_{1,2}(t)$$

Usando las dos funciones 3.12 y 3.13 se obtiene:

$$B_{0,3}(t) = \begin{cases} \frac{t^3}{6} & \text{si } 0 \leq t < 1 \\ \frac{1}{6}(-3t^3 + 12t^2 - 12t + 4) & \text{si } 1 \leq t < 2 \\ \frac{1}{6}(3t^3 - 24t^2 + 60t - 44) & \text{si } 2 \leq t < 3 \\ \frac{(4-t)^3}{6} & \text{si } 3 \leq t < 4 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.14)$$

Esta función se muestra en la figura 3.4.

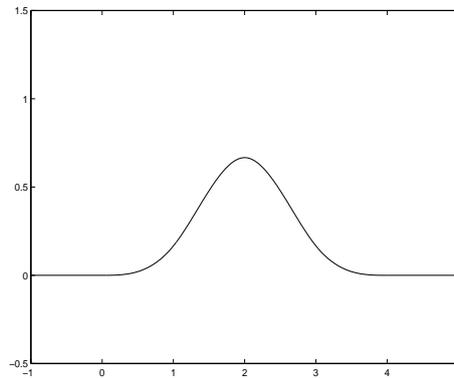


Fig. 3.4: Función base  $B_{0,3}(t)$

En la figura 3.5 se muestran los polinomios que forman esta función.

Como ya hemos citado estas funciones base son de soporte compacto (lo cual significa que la función es distinta de cero sólo en un determinado intervalo). Para una función  $B_{i,k}(t)$  su soporte es el intervalo  $[i, i + k + 1]$ . Esta es una propiedad

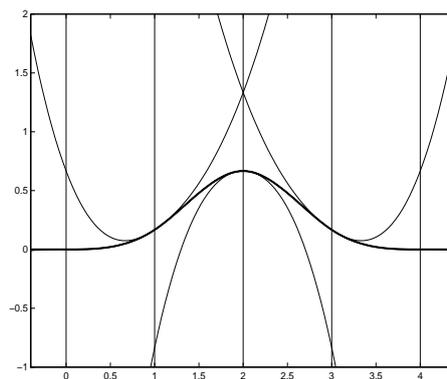


Fig. 3.5: Partes de la función base  $B_{0,3}(t)$

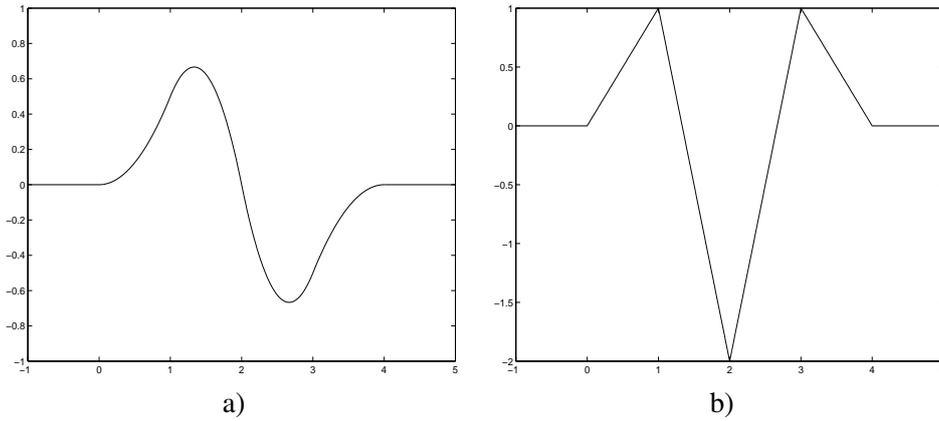


Fig. 3.6: Derivada de la función  $B_{0,3}$ . a)  $B'_{0,3}$  y b)  $B''_{0,3}$

importante y haremos uso de ella más adelante.

A partir de  $B_{i,k}(t)$  en esta forma es fácil calcular la primera y segunda derivadas:

$$B'_{0,3}(t) = \begin{cases} \frac{t^2}{2} & \text{si } 0 \leq t < 1 \\ \frac{1}{6}(-9t^2 + 24t - 12) & \text{si } 1 \leq t < 2 \\ \frac{1}{6}(9t^2 - 48t + 60) & \text{si } 2 \leq t < 3 \\ \frac{(4-t)^2}{2} & \text{si } 3 \leq t < 4 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.15)$$

$$B''_{0,3}(t) = \begin{cases} t & \text{si } 0 \leq t < 1 \\ \frac{1}{6}(-18t + 24) & \text{si } 1 \leq t < 2 \\ \frac{1}{6}(18t - 48) & \text{si } 2 \leq t < 3 \\ 4 - t & \text{si } 3 \leq t < 4 \\ 0 & \text{en otro caso.} \end{cases} \quad (3.16)$$

Las gráficas de la primera y segunda derivada se muestran en la figura 3.7

### Representación matricial.

En lugar de la representación mostrada en la ecuación 3.3 se puede construir una representación matricial. Sea la función  $B_{0,3,\delta}(t)$  la función base 0 de grado 3 y que ha sido muestreada con un espaciado  $\delta$  en el intervalo  $[i, i + 4]$ , esto significa que nos quedaremos con  $\frac{4}{\delta} + 1$  puntos de la función. Sea el conjunto de muestras

$a_0, a_1, \dots, a_{\frac{4}{\delta}+1}$  con  $a_0 = 0$  y  $a_{\frac{4}{\delta}+1} = 0$ . En la figura 3.7 está la función base muestreada con un espaciado de 0,2.

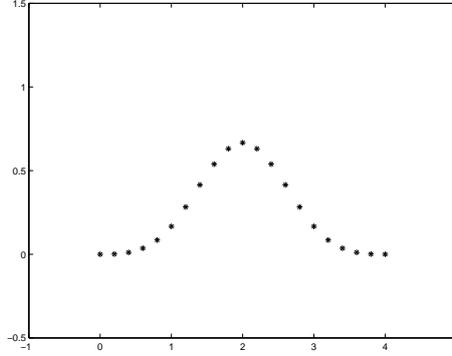


Fig. 3.7: Muestras de la función base  $B_{0,3}(t)$  con  $\delta = 0,2$

Supongamos que queremos aproximar el conjunto de puntos

$$(x_1, y_1), (x_1, y_1), \dots, (x_n, y_n)$$

y que este conjunto de puntos describen un contorno cerrado. Para dar cuenta de la periodicidad de la curva organizamos los puntos del siguiente modo:

$$(x_{n-2}, y_{n-1}), (x_{n-1}, y_{n-1}), (x_n, y_n), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

realizando un cambio de índice se reorganizan para obtener  $(x_j, y_j \ j = 1, \dots, n+3)$ . El polinomio para la coordenada  $x$  será:

$$Q_x = \sum_{j=1}^{n+3} x_j B_{j,3}(t)$$

y para la coordenada  $y$ :

$$Q_y = \sum_{j=1}^{n+3} y_j B_{j,3}(t)$$

Puesto que la curva interpolada se obtiene como el producto de cada punto de control por la función base desplazada una unidad se pueden escribir estas ecuaciones en forma matricial:

$$\mathbf{Q}_x = \mathbf{A}\mathbf{x} \quad \mathbf{Q}_y = \mathbf{A}\mathbf{y} \quad (3.17)$$

Si llamamos  $\mathbf{A}'$  a la matriz de coeficientes de la primera derivada y  $\mathbf{A}''$  a la matriz de coeficiente de la segunda derivada las curvas de la primera y segunda derivada de la curva aproximadora son:

$$\mathbf{Q}'_{\mathbf{x}} = \mathbf{A}'\mathbf{x} \quad \mathbf{Q}'_{\mathbf{y}} = \mathbf{A}'\mathbf{y} \quad (3.18)$$

$$\mathbf{Q}''_{\mathbf{x}} = \mathbf{A}''\mathbf{x} \quad \mathbf{Q}''_{\mathbf{y}} = \mathbf{A}''\mathbf{y} \quad (3.19)$$

De este modo se observa claramente que si movemos un punto de control lo único que tenemos que hacer para obtener la nueva curva son dos multiplicaciones de matrices.

### 3.2.2. Parametrización del snake usando B-splines

Ahora se considerará que la curva no está formada por puntos aislados sino que esos puntos son los puntos de control de un *B-spline*, por tanto la curva estará definida (según la ecuación 3.3 mediante:

$$v(s) = (x(s), y(s)) = \left( \sum_{i=1}^n x_i B_{i,3}(s), \sum_{i=1}^n y_i B_{i,3}(s) \right)$$

ó según la ecuación 3.17

$$v_j = (x_j, y_j) = (\mathbf{Ax}, \mathbf{Ay}) \quad j = 1, \dots, j = \frac{n}{\delta}$$

donde  $n$  es el número de puntos de control, y  $\delta$  es el espaciado de muestreo.

En este caso la energía del *snake* será, usando las ecuaciones 3.17, 3.18 y 3.19

$$E_{snake} = \alpha \{ (\mathbf{A}'\mathbf{x})^2 + (\mathbf{A}'\mathbf{y})^2 \} + \beta \{ (\mathbf{A}''\mathbf{x})^2 + (\mathbf{A}''\mathbf{y})^2 \} + E_{ext}(\mathbf{Ax}, \mathbf{Ay}) \quad (3.20)$$

Con esto se ha finalizado el capítulo de la representación de contornos activos, donde se han presentado dos posibles representaciones. En el siguiente capítulo se comentará como minimizar estas ecuaciones y se presentarán diferentes alternativas para la energía externa.



## 4. ENERGÍA EXTERNA. MINIMIZACIÓN DE LA ENERGÍA

En este capítulo se estudiarán dos posibles elecciones de energía externa:

- $-|\nabla I * G_\sigma|$  y
- la distancia *chamfer*

además se verán las distintas posibilidades a la hora de abordar la minimización de la energía, concretamente se analizarán en detalle la solución numérica (directamente a partir de la ecuación de Euler-Lagrange) y un algoritmo de minimización voraz.

### 4.1. Energía externa

El objetivo del funcional de energía externa es atraer el snake hacia extremos de intensidad, bordes, o cualquier otra propiedad de la imagen que se defina. La energía externa será una función definida sobre la imagen y se calculará mediante el procesamiento de la misma. Habitualmente se interpretan los mínimos locales de la energía externa como atractores del snake.

En el trabajo de Kass [11] la energía externa utilizada para detectar los bordes es el módulo de la convolución de una máscara gaussiana de varianza  $\sigma$  con el gradiente de la imagen,  $-|G_\sigma * \nabla I(x, y)|$ . El problema que tiene este funcional de energía es que si el contorno inicial está alejado de los bordes del objeto, la energía en esas zonas será cero y el snake se verá influenciado únicamente por la energía interna.

Otra posible elección para la energía externa es el uso de la distancia chamfer o transformación de distancia. La distancia chamfer se define como la distancia que hay desde cualquier punto del dominio de definición de la imagen al borde más cercano. De este modo se consigue que la energía externa sea distinta de cero en todo punto de la imagen salvo en los bordes.

El método para el cálculo de la distancia chamfer es el propuesto por Borgefors [4] y se describe a continuación.

Se parte de una imagen de bordes en la que los puntos "borde" se fijan a 0 y los demás puntos se fijan a  $\infty$  (a efectos prácticos se pondría un valor suficientemente

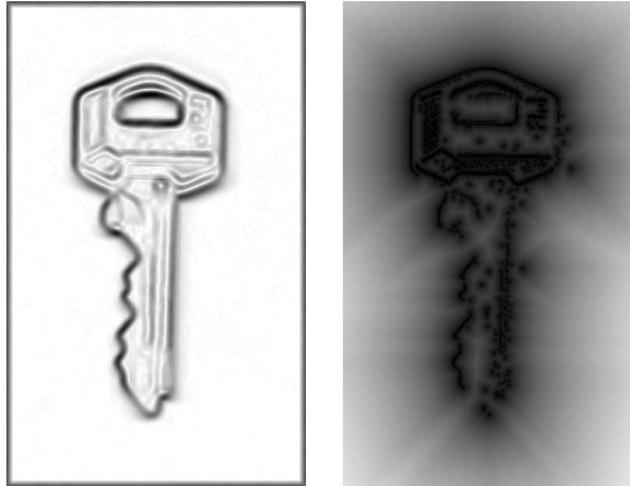


Fig. 4.1:  $-|G_\sigma \star \nabla I(x, y)|$  y chamfer

elevado). El algoritmo que obtiene la transformación de distancia parte de esta imagen transformada. Se realizan dos pasadas sobre la imagen, primero hacia adelante de izquierda a derecha y de arriba a abajo; y a continuación otra hacia atrás de derecha a izquierda y de abajo hacia arriba, según el siguiente esquema:

*Hacia adelante:*

para  $i = 2, \dots$ , filas hacer

para  $j = 2, \dots$ , columnas hacer

$$v_{i,j} = \min(v_{i-1,j-1} + 4, v_{i-1,j} + 3, v_{i-1,j+1} + 4, v_{i,j-1} + 3, v_{i,j})$$

*Hacia atrás:*

para  $i = \text{filas} - 1, \dots, 1$  hacer

para  $j = \text{columnas} - 1, \dots, 1$  hacer

$$v_{i,j} = \min(v_{i+1,j-1} + 4, v_{i+1,j} + 3, v_{i+1,j+1} + 4, v_{i+1,j} + 3, v_{i,j})$$

En la figura 4.1 se puede ver como la distancia chamfer disminuye su valor gradualmente hacia los bordes desde cualquier posición mientras que para la convolución con el gradiente en posiciones lejanas a los bordes la energía tiene un valor casi constante.

También se podría pensar en usar otra energía interna que según el tipo de imágenes pueden dar buenos resultados [9].

## 4.2. Minimización de la energía

Una vez elegida la representación del snake y las energías a utilizar, se debe partir de un contorno inicial y llegar hasta un contorno final mediante la minimización de la energía. Se pueden aplicar muchos métodos para minimizar la energía, por citar algunos: resolución numérica [11], programación dinámica [1], algoritmos voraces [18] [10] [8], y métodos estadísticos. En esta sección se verán tres métodos para minimizar la energía del contorno activo para los dos métodos de representación del mismo: **snake** (puntos aislados) y **B-snake** (puntos aproximados mediante un B-spline). Los tres métodos son los siguientes:

- Método numérico propuesto por Kass para el snake.
- Método de sobrerrelajación sucesiva (SOR) para el snake.
- Algoritmo voraz para el snake y el B-snake.

### 4.2.1. Stiffness matrix

Partimos de la ecuación

$$E_{snake}(v(s)) = \int_{s_a}^{s_b} \alpha(s)|v_s|^2 + \beta(s)|v_{ss}|^2 + E_{ext}(v(s))ds \quad (4.1)$$

Este funcional es del tipo

$$\mathcal{I}(v) = \int_{s_a}^{s_b} F(s, v, v_s, v_{ss})ds$$

si una función  $v(s)$  minimiza la integral  $\mathcal{I}(v)$  entonces debe cumplir la ecuación de Euler-Lagrange (que es una condición necesaria pero no suficiente), que para este funcional es la siguiente (ver Apendice A):

$$F_v - \frac{d}{ds}F_{v_s} + \frac{d^2}{ds^2}F_{v_{ss}} = 0 \quad (4.2)$$

En el funcional  $\mathcal{I}(v)$  solo hay un término que depende de  $v$  que es el de energía externa, otro que solamente depende de  $v_s$  que es el de elasticidad y otro que sólo depende de  $v_{ss}$  que es el de rigidez. Por tanto se puede reescribir la ecuación 4.2 como

$$\frac{d}{dv}E_{ext}(v) - \frac{d}{ds}(\alpha(v_i - v_{i-1})) + \frac{d^2}{ds^2}(\beta(v_{i-1} - 2v_i + v_{i+1})) = 0$$

donde hemos asumido que  $\alpha$  y  $\beta$  son constantes. Realizando la derivación discreta y reorganizando los términos en función de las coordenadas (ver Apéndice B) se obtienen dos ecuaciones matriciales de la forma

$$\mathbf{K}\mathbf{x} = \mathbf{f}_x \quad \mathbf{K}\mathbf{y} = \mathbf{f}_y \quad (4.3)$$

donde  $\mathbf{K}$  (que es conocida en la literatura como *stiffness matrix*) es una matriz pentadiagonal y dispersa relacionada con la discretización. Por tanto se ha reducido el problema a resolver un par de sistemas de ecuaciones lineales. A continuación veremos dos métodos para la resolución de esos dos sistemas.

#### 4.2.2. Método de Kass

Este método se basa en calcular una solución de las ecuaciones 4.3 mediante un método iterativo. Concretamente se iguala la parte izquierda de las ecuaciones 4.3 al producto de una constante por la derivada discreta respecto al tiempo de las coordenadas. Con ello las ecuaciones son:

$$\begin{aligned} \mathbf{K}\mathbf{x}_t + \mathbf{f}_x(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= -\gamma(\mathbf{x}_t - \mathbf{x}_{t-1}) \\ \mathbf{K}\mathbf{y}_t + \mathbf{f}_y(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= -\gamma(\mathbf{y}_t - \mathbf{y}_{t-1}) \end{aligned} \quad (4.4)$$

En el equilibrio  $\mathbf{x}_t = \mathbf{x}_{t-1}$  y  $\mathbf{y}_t = \mathbf{y}_{t-1}$ , con lo que se habrá conseguido la solución a las ecuaciones 4.3. Despejando  $\mathbf{x}_t$  e  $\mathbf{y}_t$ :

$$\begin{aligned} \mathbf{x}_t &= (\mathbf{K} + \gamma\mathbf{I})^{-1}(\mathbf{x}_{t-1} - \mathbf{f}_x(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})) \\ \mathbf{y}_t &= (\mathbf{K} + \gamma\mathbf{I})^{-1}(\mathbf{y}_{t-1} - \mathbf{f}_y(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})) \end{aligned} \quad (4.5)$$

donde  $\mathbf{I}$  es la matriz identidad. Puesto que se supondrá que la matriz  $\mathbf{K} + \gamma\mathbf{I}$  es constante (ver Apéndice B) solo será necesario calcular su inversa una vez.

#### 4.2.3. Solución por sobrerelajación sucesiva.

Partiendo de nuevo de las ecuaciones 4.3 construimos otro par de ecuaciones equivalentes usando una matriz  $\mathbf{M}$  [14]:

$$\begin{aligned} \mathbf{M}\mathbf{x} &= (\mathbf{M} - \mathbf{K})\mathbf{x} + \mathbf{f}_x \\ \mathbf{M}\mathbf{y} &= (\mathbf{M} - \mathbf{K})\mathbf{y} + \mathbf{f}_y \end{aligned}$$

Esta ecuación se resuelve mediante sustituciones sucesivas. Se comienza con un valor inicial (que vendrá dado por el contorno inicial)  $\mathbf{x}_0$  y  $\mathbf{y}_0$  y cada valor  $\mathbf{x}_{k+1}$  y  $\mathbf{y}_{k+1}$  se calculará del siguiente modo:

$$\begin{aligned} \mathbf{M}\mathbf{x}_{k+1} &= (\mathbf{M} - \mathbf{K})\mathbf{x}_k + \mathbf{f}_{\mathbf{x}_k} \\ \mathbf{M}\mathbf{y}_{k+1} &= (\mathbf{M} - \mathbf{K})\mathbf{y}_k + \mathbf{f}_{\mathbf{y}_k} \end{aligned}$$

Hay tres posibles elecciones de la matriz  $\mathbf{M}$  que funcionan correctamente y que se derivan directamente de  $\mathbf{K}$ :

1.  $\mathbf{M}$  = parte diagonal de  $\mathbf{K}$  (método de Jacobi).
2.  $\mathbf{M}$  = parte triangular de  $\mathbf{K}$  (método de Gauss-Seidel).
3.  $\mathbf{K}$  = combinación de 1) y 2) (sobre-relajación sucesiva).

La convergencia de los vectores  $\mathbf{x}_0, \mathbf{x}_1, \dots$  no se puede asegurar. Pero se puede afirmar lo siguiente: si se aproximan a un límite  $x_\infty$ , entonces ese límite es la solución correcta del sistema 4.3. Además de la convergencia, es igualmente importante la velocidad de convergencia. Si  $\mathbf{x}_k$  converge rápidamente a  $\mathbf{K}^{-1}\mathbf{f}_x$ , entonces el método iterativo es una alternativa a la eliminación gaussiana, no dará la respuesta exacta en un tiempo finito, pero puede dar una buena aproximación mucho antes. La convergencia o divergencia de los métodos iterativos depende de  $\mathbf{M}$  y de  $\mathbf{K}$ . El método de sobre-relajación sucesiva se basa en elegir  $\mathbf{M}$  como la parte triangular inferior de la matriz  $\mathbf{K}$  y dividir la diagonal por un factor  $w$  con  $1 < w < 2$  para asegurar la convergencia.

#### 4.2.4. Algoritmos voraces

Ahora cambia radicalmente el enfoque, ya que pasa a ser computacional en lugar de numérico. Un algoritmo voraz es aquel en el que se minimiza localmente la función objetivo y la solución encontrada en cada uno de los pasos locales pasa a formar parte del conjunto solución. En el problema que nos ocupa, tanto para el snake como para el B-snake la energía depende localmente del punto considerado. En efecto, para el snake la energía en un punto  $v_i$  depende de tres puntos de control  $v_{i-1}, v_i, v_{i+1}$  y para el B-snake, como ya se comentó en el capítulo anterior sobre representación, los B-splines son de soporte compacto por lo que dependen localmente de cada punto de control.

Por tanto para las dos posibles representaciones cada punto influye localmente por lo que se pueden considerar por separado, calcular su energía uno a uno y se puede minimizar la energía total mediante una estrategia voraz.

El método es el siguiente: minimizar la energía en una vecindad de cada punto de control, desplazar el punto a la posición de mínima energía y pasar a considerar

el siguiente punto. El proceso se repite mientras no se cumpla una cierta condición de final. Desde un punto de vista más formal, sea el conjunto de puntos de control  $\mathcal{V}^{(k)} = v_i^{(k)} \quad i = 1, \dots, N$  tras la iteración k-ésima. La energía total en esa iteración será:

$$E_{snake}^{(k)} = E_{snake}(\mathcal{V}^{(k)}) = \sum_{i=1}^N e(v_i^{(k)})$$

donde  $e(v_i^{(k)})$  es la energía calculada en la zona de influencia del i-ésimo punto de control (ya sea una porción de curva si se se trata del B-snake o de tres puntos si se trata del snake). Supongamos que sobre cada punto de control colocamos una ventana  $\Omega(v_i^{(k)})$  centrada en  $v_i^{(k)}$ , en la cual cualquier punto  $v \in \Omega$  es candidato para el cálculo de la energía. El punto  $v_i^{(k)}$  se mueve a la posición  $v_i^{(k+1)}$  según la siguiente regla de actualización:

$$v_i^{(k+1)} = \underset{v \in \Omega(v_i^{(k)})}{\operatorname{argmin}} [e_i(v)]$$

para  $i = 1, \dots, n$ . Tras aplicar la regla de actualización anterior, la energía obtenida es menor que la original, por tanto  $E_{snake}^{(k)}$  es monótonamente decreciente. Una vez obtenida la posición de mínima energía el punto de control se actualiza a esa posición y se continúa con el siguiente punto de control. La situación se muestra en la figura 4.2.

En lugar de la ventana cuadrada mostrada en la figura anterior, en este trabajo se ha usado un espacio de búsqueda lineal, en la perpendicular a la tangente en el punto considerado. La situación se muestra en la figura 4.3.

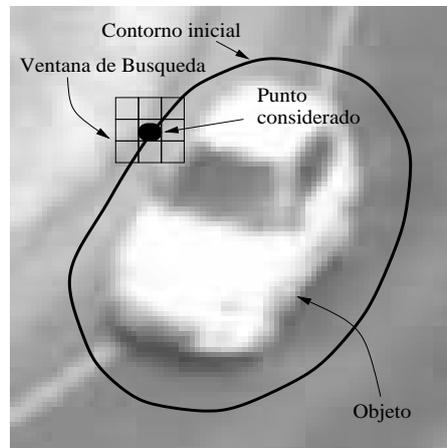


Fig. 4.2: Ventana de búsqueda cuadrada.

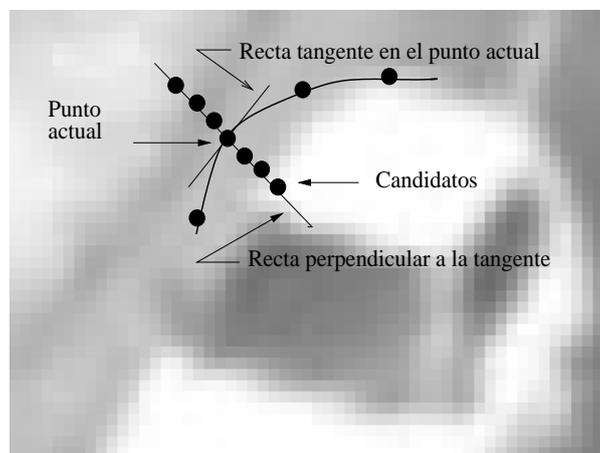


Fig. 4.3: Ventana de búsqueda lineal



## 5. EXPERIMENTOS Y RESULTADOS.

En este capítulo se presentarán los experimentos realizados para probar los modelos, representaciones y minimizaciones descritas en los capítulos previos.

### 5.1. Experimentos

Los modelos se probarán sobre 6 imágenes: cinco de ellas de laboratorio y una escena real. Estas imágenes se muestran en las figuras 5.1, 5.2 y 5.3:

Como puede observarse, las imágenes de la figura 5.1 han sido etiquetadas como Lab1, Lab2, Lab3 y Lab4. Lab1 presenta zonas de diversa curvatura suave, Lab2 está formada por líneas discontinuas, Lab3 es la más sencilla de todas ya que consta de segmentos rectos unidos por arcos de curvatura suave y Lab4 presenta cuatro esquinas con ángulos casi rectos.

La imagen de la figura 5.2 es una llave y es quizá la imagen más complicada de todas ya que presenta muchas zonas con alta curvatura y es un objeto con anchura variable.

La imagen de la figura 5.3 es una escena real de tráfico en la que aparece un vehículo circulando por una curva y es la única imagen que contiene más de un objeto, en este caso se pretende ver si el modelo es capaz de discriminar correctamente el vehículo.

Con cada una de estas imágenes se realizarán 8 pruebas que se describen a continuación:

- Método de minimización de Kass con  $-|G_\sigma * \nabla I|$ . Llamaremos a este experimento **Kass-conv**.
- Método de minimización de Kass con distancia chamfer. Llamaremos a este experimento **Kass-ch**.
- Método de minimización mediante SOR con  $-|G_\sigma * \nabla I|$ . Llamaremos a este experimento **SOR-conv**.
- Método de minimización mediante SOR con distancia chamfer. Llamaremos a este experimento **SOR-ch**.

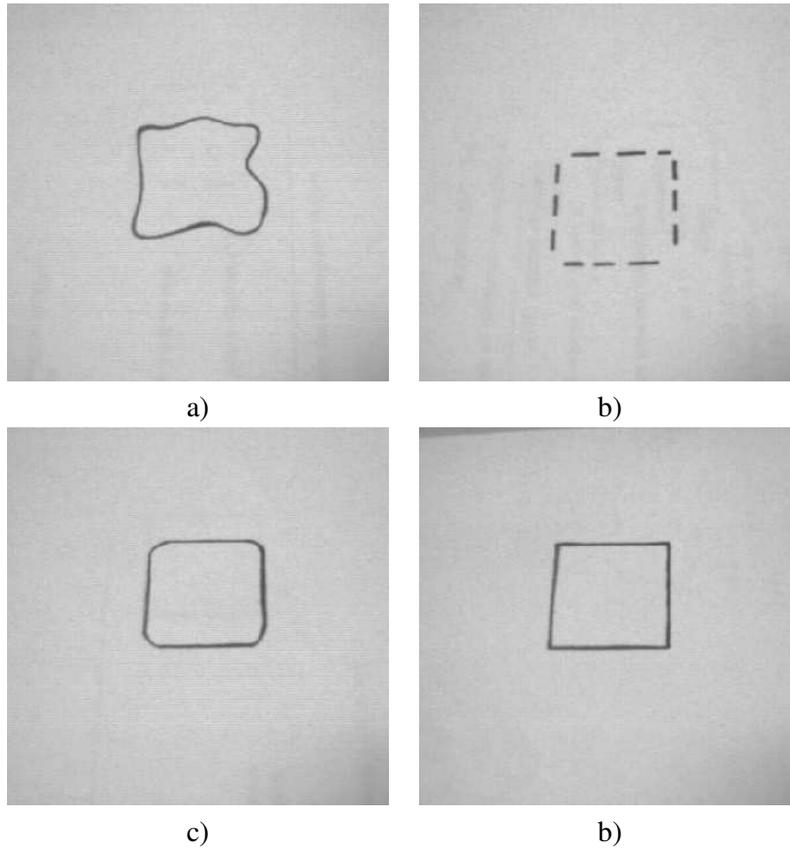


Fig. 5.1: Imágenes sintéticas. a) Lab1, b) Lab2, c) Lab3 y d) Lab4



*Fig. 5.2: Imagen real: llave*



*Fig. 5.3: Imagen real: coche*

- Método de minimización voraz sobre puntos aislados con  $-|G_\sigma * \nabla I|$ . Llamaremos a este experimento **snake-conv**.
- Método de minimización voraz sobre puntos aislados con distancia chamfer. Llamaremos a este experimento **snake-ch**.
- Método de minimización voraz sobre un B-spline con  $-|G_\sigma * \nabla I|$ . Llamaremos a este experimento **bsnake-conv**.
- Método de minimización voraz sobre un B-spline usando distancia chamfer. Llamaremos a este experimento **bsnake-ch**.

Los experimentos se han realizado sobre un Pentium 100 con 40Mb de RAM y el lenguaje de programación usado ha sido Matlab.

Además de obtener los contornos finales se ha medido el tiempo que ha necesitado cada método en minimizar su correspondiente ecuación. Estos tiempos son orientativos ya que Matlab no es precisamente un lenguaje rápido si se tienen que utilizar bucles.

Para cada experimento se han elegido heurísticamente los parámetros que hay que fijar de modo que se muestra el mejor contorno final que obtiene cada método. Los contornos iniciales se han fijado a mano ya que este no es el objetivo del presente trabajo, aunque existe la posibilidad de obtenerlos automáticamente. En la siguiente sección se muestran los resultados obtenidos.

## 5.2. Resultados experimentales.

### 5.2.1. Resultados sobre las imágenes sintéticas.

En la figura 5.4 se muestran los resultados sobre Lab1 usando como energía externa  $-|G_\sigma * \nabla I|$ .

En la tabla 5.1 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-conv	SOR-conv	snake-conv	bsnake-conv
Lab1	11.86	7.64	19.58	94.7

Tab. 5.1: Tiempos para Lab1 con  $-|G_\sigma * \nabla I|$ .

En la figura 5.5 se muestran los resultados sobre Lab1 usando como energía externa la distancia chamfer.

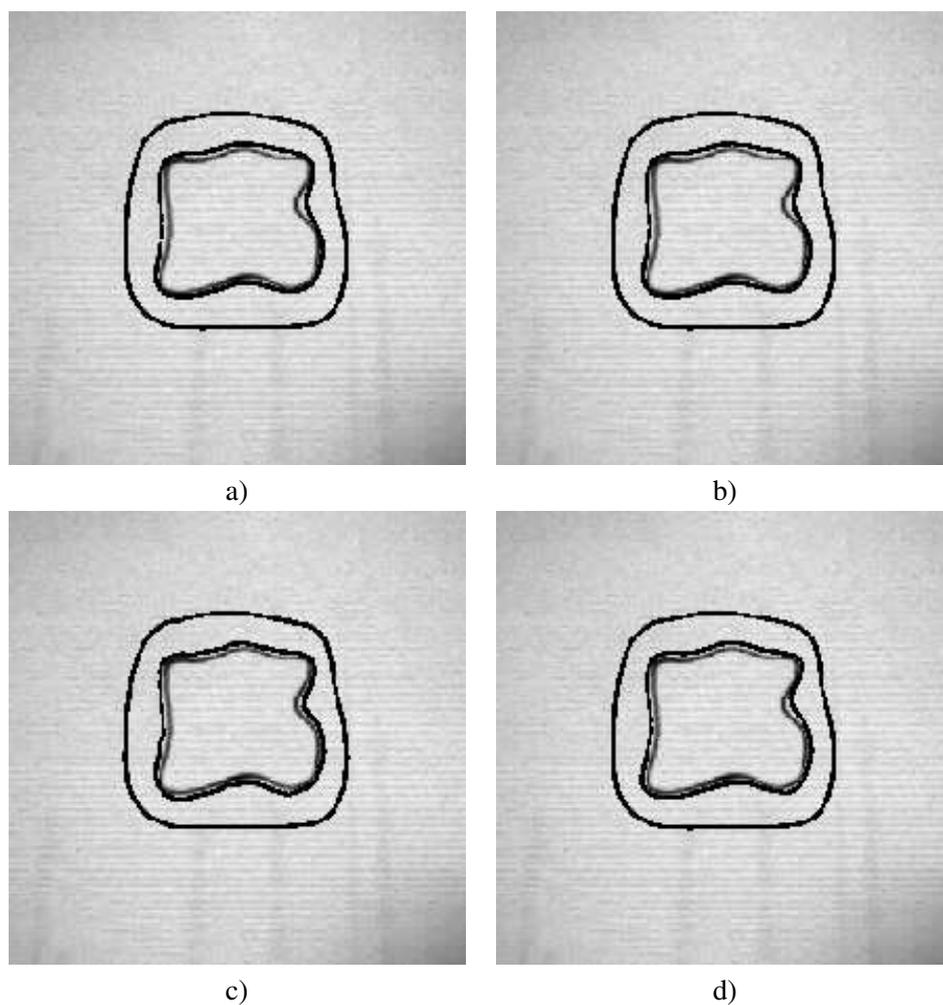


Fig. 5.4: Lab1 con  $-|G_\sigma * \nabla I|$ . a) Kass-conv, b) SOR-conv, c) snake-conv y d) bsnake-conv.

En la tabla 5.2 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-ch	SOR-ch	snake-ch	bsnake-ch
Lab1	1.69	0.79	9.07	39.5

Tab. 5.2: Tiempos para Lab1 con distancia chamfer.

En la figura 5.6 se muestran los resultados sobre Lab2 usando como energía externa  $-|G_\sigma * \nabla I|$ .

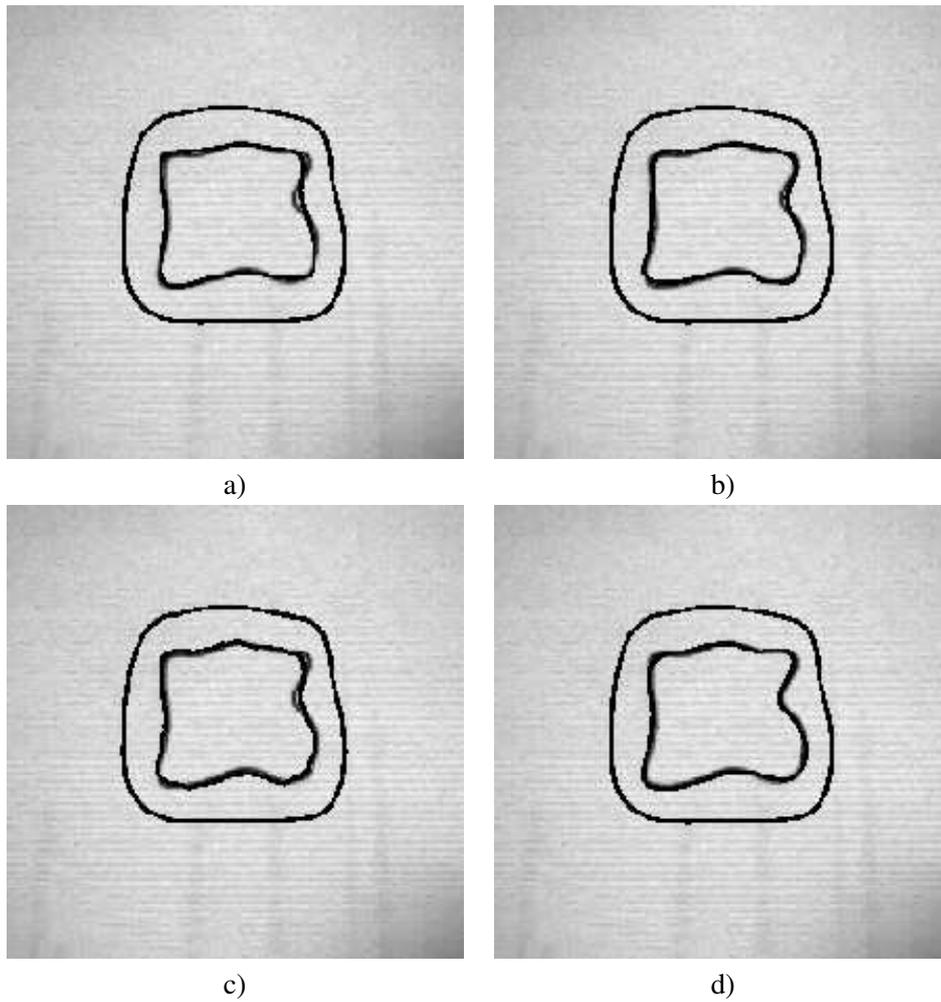


Fig. 5.5: Lab1 con distancia chamfer .a) Kass-ch, b) SOR-ch, c) snake-ch y d) bsnaek-ch.

En la tabla 5.3 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-conv	SOR-conv	snake-conv	bsnaek-conv
Lab2	7.99	4.35	19.38	98.9

Tab. 5.3: Tiempos para Lab2 con  $-|G_\sigma * \nabla I|$ .

En la figura 5.7 se muestran los resultados sobre Lab2 usando como energía externa la distancia chamfer.

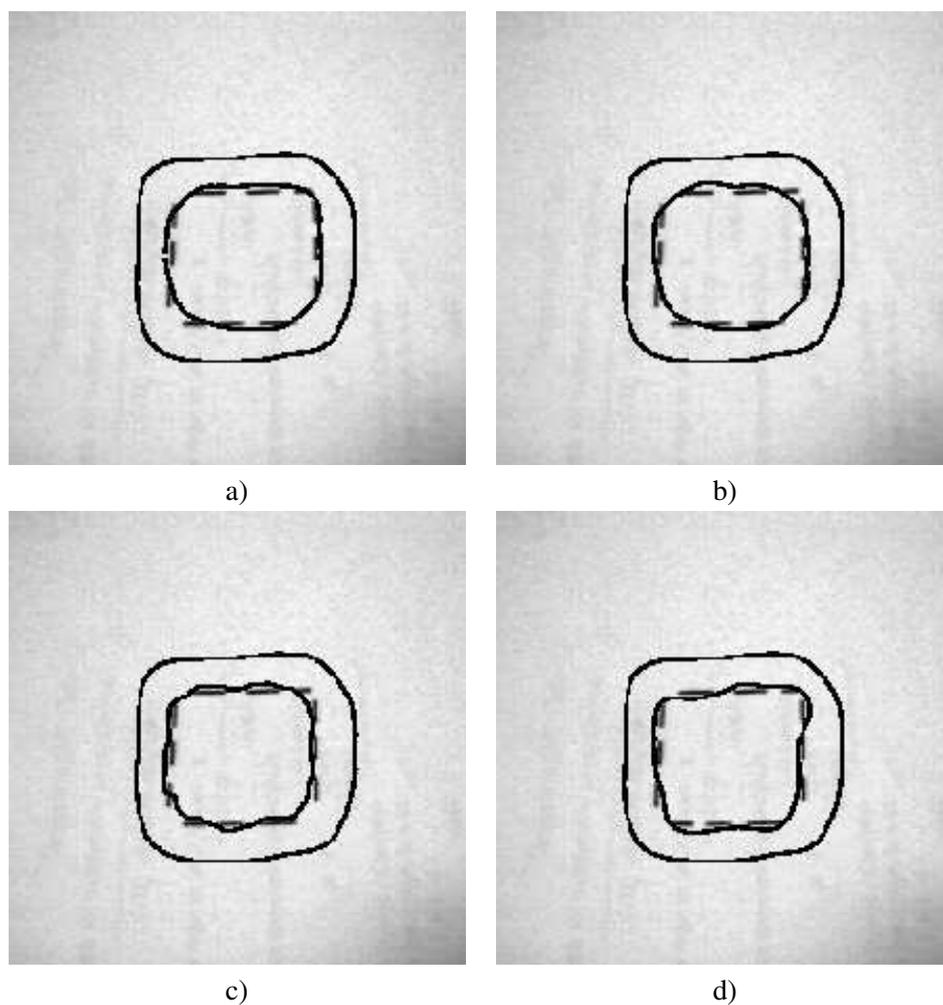


Fig. 5.6: Lab2 con  $-|G_\sigma * \nabla I|$ . a) Kass-conv, b) SOR-conv, c) snake-conv y d) bsnake-conv.

En la tabla 5.4 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-ch	SOR-ch	snake-ch	bsnake-ch
Lab2	1.18	0.43	8.95	42.59

Tab. 5.4: Tiempos para Lab2 con distancia chamfer.

En la figura 5.8 se muestran los resultados sobre Lab3 usando como energía externa  $-|G_\sigma * \nabla I|$ .

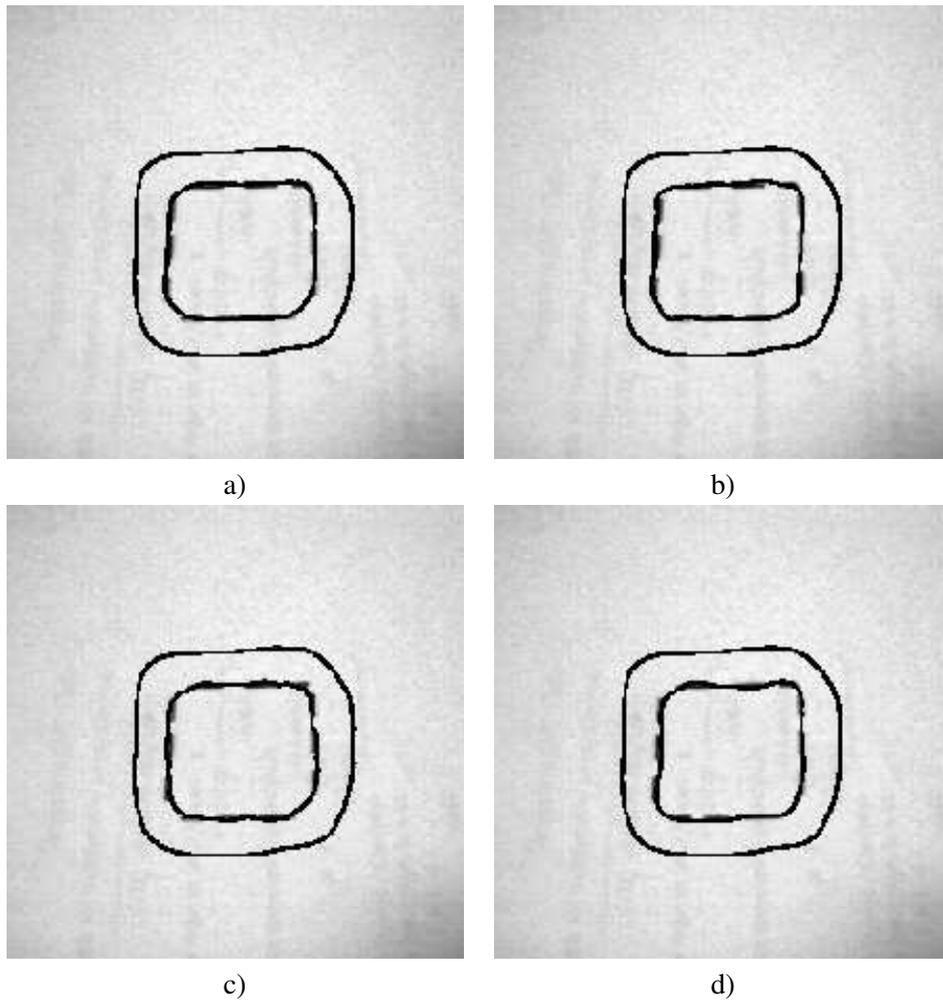


Fig. 5.7: Lab2 con distancia chamfer .a) Kass-ch, b) SOR-ch, c) snake-ch y d) bsnaek-ch.

En la tabla 5.5 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-conv	SOR-conv	snake-conv	bsnaek-conv
Lab3	10.01	5.27	12.6	99.24

Tab. 5.5: Tiempos para Lab3 con  $-|G_\sigma * \nabla I|$ .

En la figura 5.9 se muestran los resultados sobre Lab3 usando como energía externa la distancia chamfer.

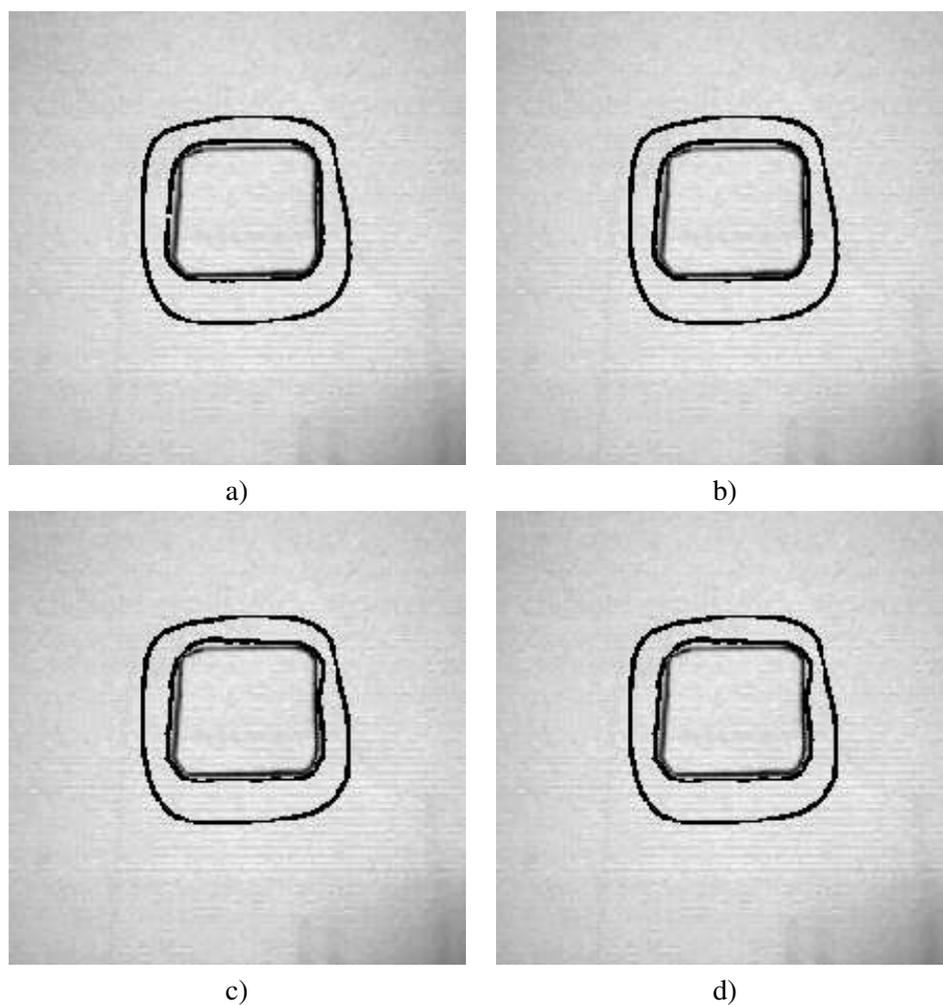


Fig. 5.8: Lab3 con  $-|G_\sigma * \nabla I|$ . a) Kass-conv, b) SOR-conv, c) snake-conv y d) bsnake-conv.

En la tabla 5.6 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-ch	SOR-ch	snake-ch	bsnake-ch
Lab3	0.77	0.5	9.19	42.49

Tab. 5.6: Tiempos para Lab3 con distancia chamfer.

En la figura 5.10 se muestran los resultados sobre Lab4 usando como energía externa  $-|G_\sigma * \nabla I|$ .

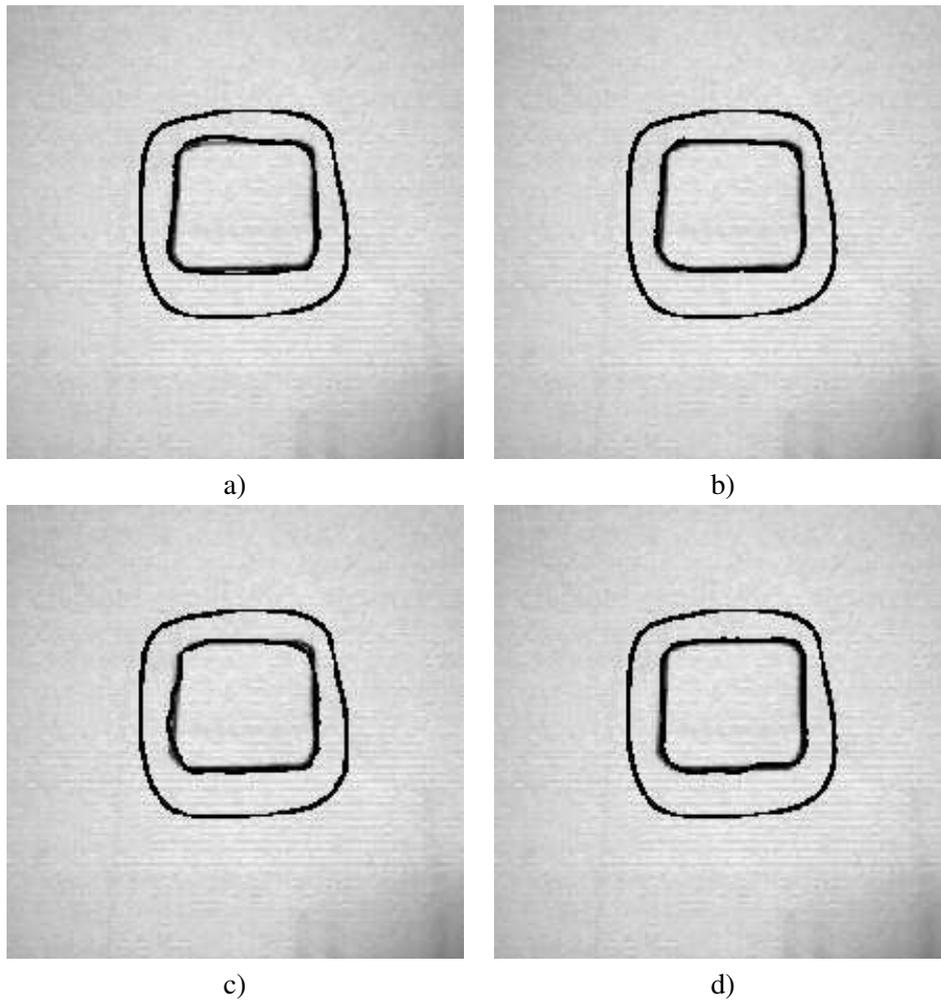


Fig. 5.9: Lab3 con distancia chamfer .a) Kass-ch, b) SOR-ch, c) snake-ch y d) bsnaek-ch.

En la tabla 5.7 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-conv	SOR-conv	snake-conv	bsnaek-conv
Lab4	9.17	6.11	11.8	86.7

Tab. 5.7: Tiempos para Lab4 con  $-|G_\sigma * \nabla I|$ .

En la figura 5.11 se muestran los resultados sobre Lab4 usando como energía externa la distancia chamfer.

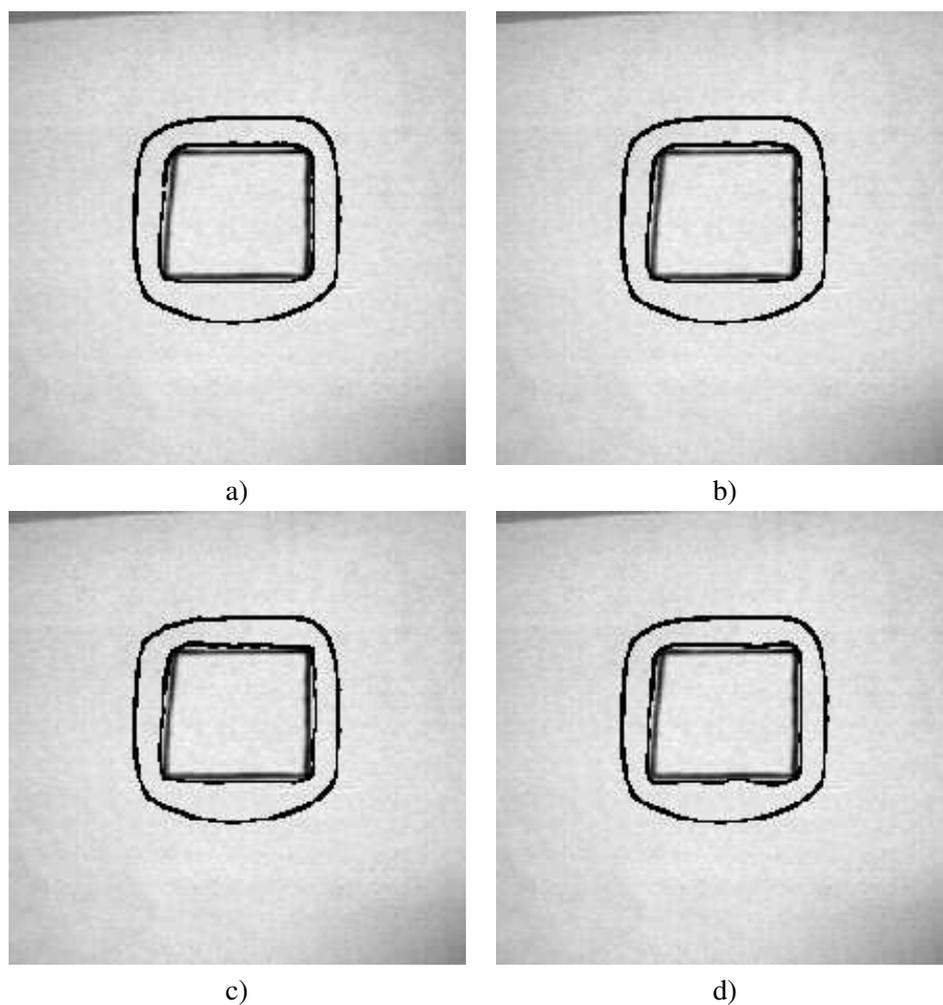


Fig. 5.10: Lab4 con  $-|G_{\sigma} * \nabla I|$ . a) Kass-conv, b) SOR-conv, c) snake-conv y d) bsnake-conv.

En la tabla 5.8 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-ch	SOR-ch	snake-ch	bsnake-ch
Lab4	1.45	0.55	8.84	47.22

Tab. 5.8: Tiempos para Lab4 con distancia chamfer.

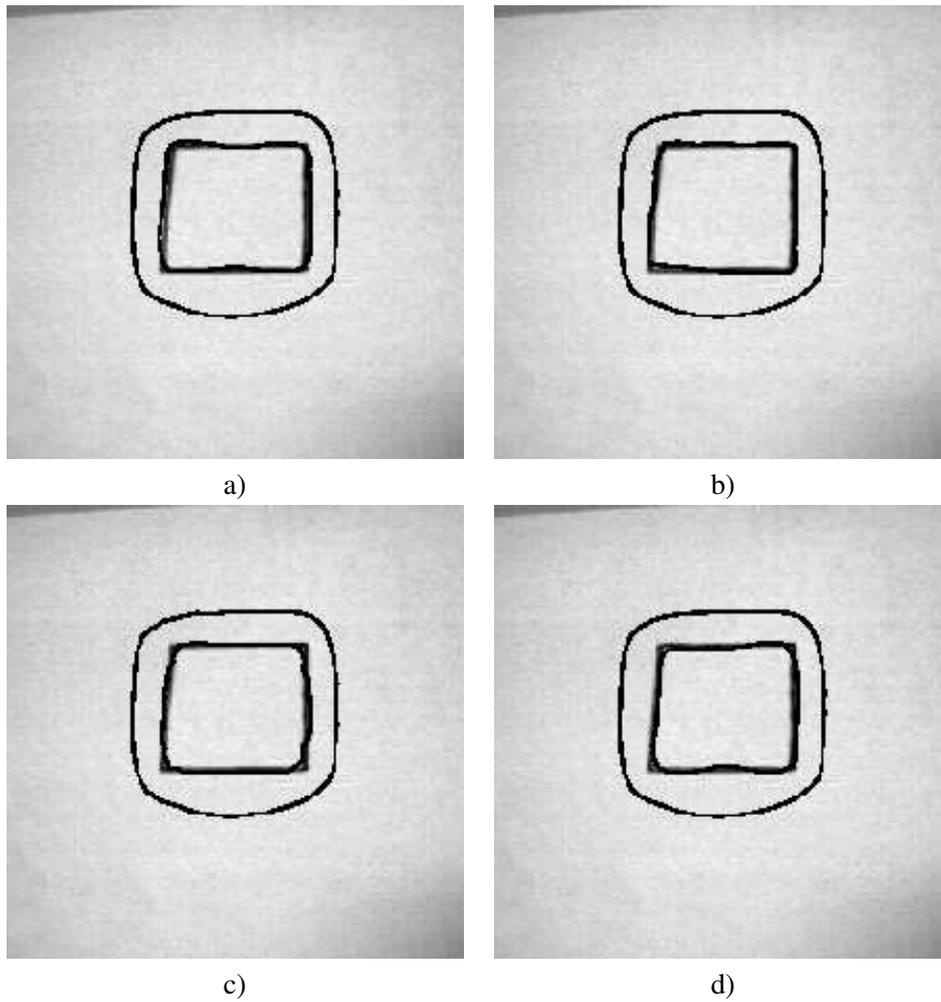


Fig. 5.11: Lab4 con distancia chamfer .a) Kass-ch, b) SOR-ch, c) snake-ch y d) bsnake-ch.

### 5.2.2. Resultados sobre las imágenes reales.

En la figura 5.12 se muestran los resultados sobre la llave, usando como energía externa  $-|G_{\sigma} * \nabla I|$ .

En la tabla 5.9 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

En la figura 5.13 se muestran los resultados sobre la llave usando como energía externa la distancia chamfer.

En la tabla 5.10 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

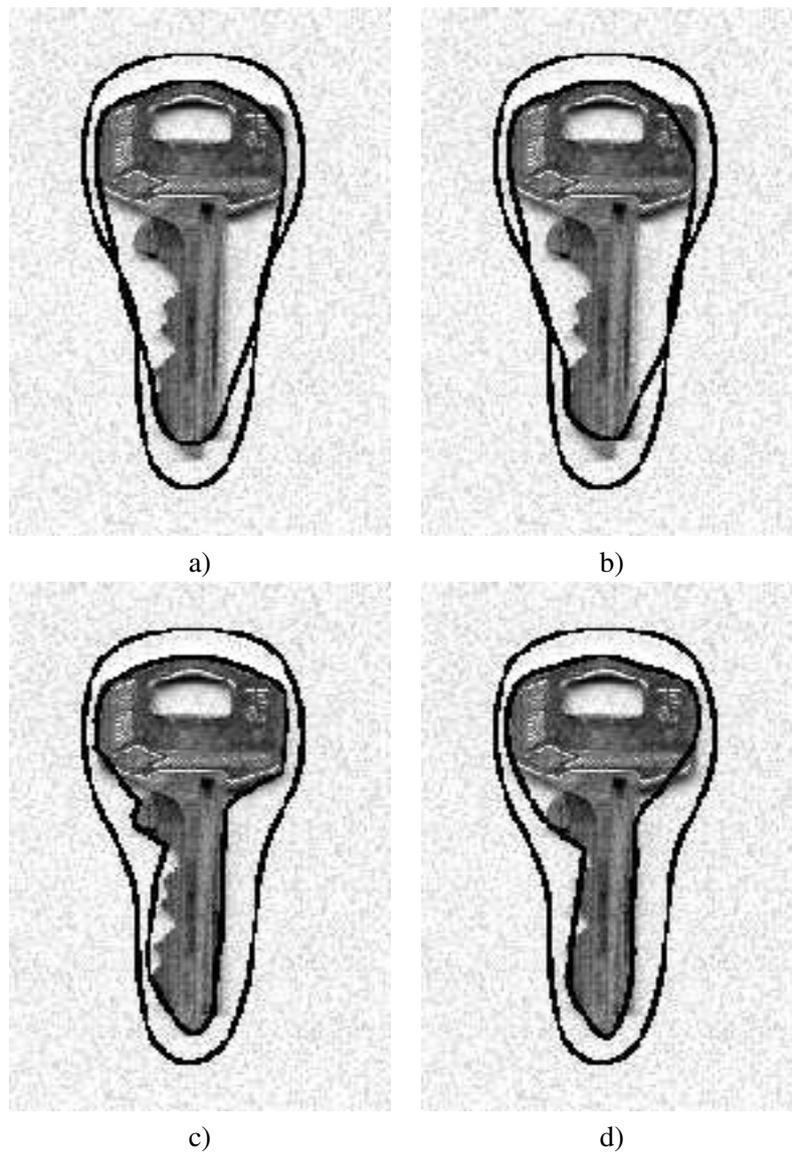


Fig. 5.12: Llave con  $-|G_\sigma * \nabla I|$ . a) Kass-conv, b) SOR-conv, c) snake-conv y d) bsnake-conv.

Imagen	Kass-conv	SOR-conv	snake-conv	bsnake-conv
Llave	3	1.56	25	82.08

Tab. 5.9: Tiempos para la llave con  $-|G_\sigma * \nabla I|$ .

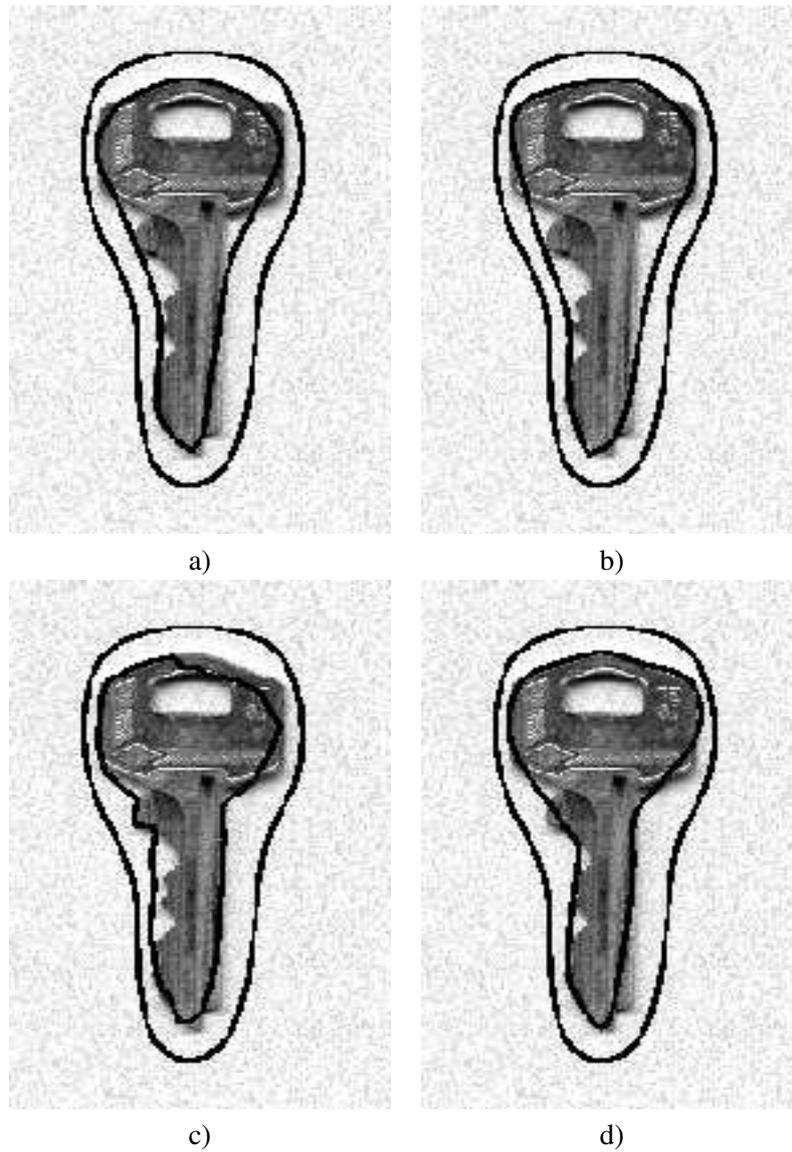


Fig. 5.13: Llave con distancia chamfer .a) Kass-ch, b) SOR-ch, c) snake-ch y d) bsnaek-ch.

Imagen	Kass-ch	SOR-ch	snake-ch	bsnaek-ch
Llave	1.15	0.5	13.8	44.0

Tab. 5.10: Tiempos para la llave con distancia chamfer.

En la figura 5.14 se muestran los resultados sobre el coche, usando como energía

externa  $-|G_\sigma * \nabla I|$ .

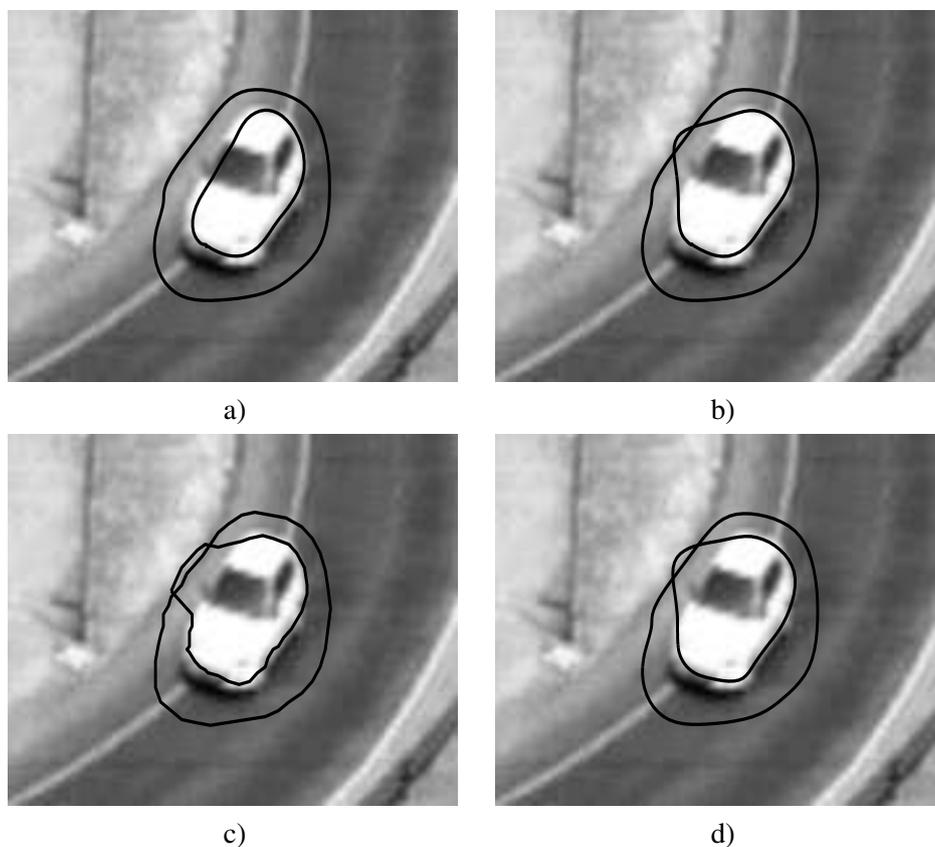


Fig. 5.14: Coche con  $-|G_\sigma * \nabla I|$ . a) Kass-conv, b) SOR-conv, c) snake-conv y d) bsnake-conv.

En la tabla 5.11 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

Imagen	Kass-conv	SOR-conv	snake-conv	bsnake-conv
Coche	8.68	1.83	9.82	61.89

Tab. 5.11: Tiempos para el coche con  $-|G_\sigma * \nabla I|$ .

En la figura 5.15 se muestran los resultados sobre el coche usando como energía externa la distancia chamfer.

En la tabla 5.12 se muestran los tiempos (en segundos) empleados por cada uno de los métodos en obtener el contorno final.

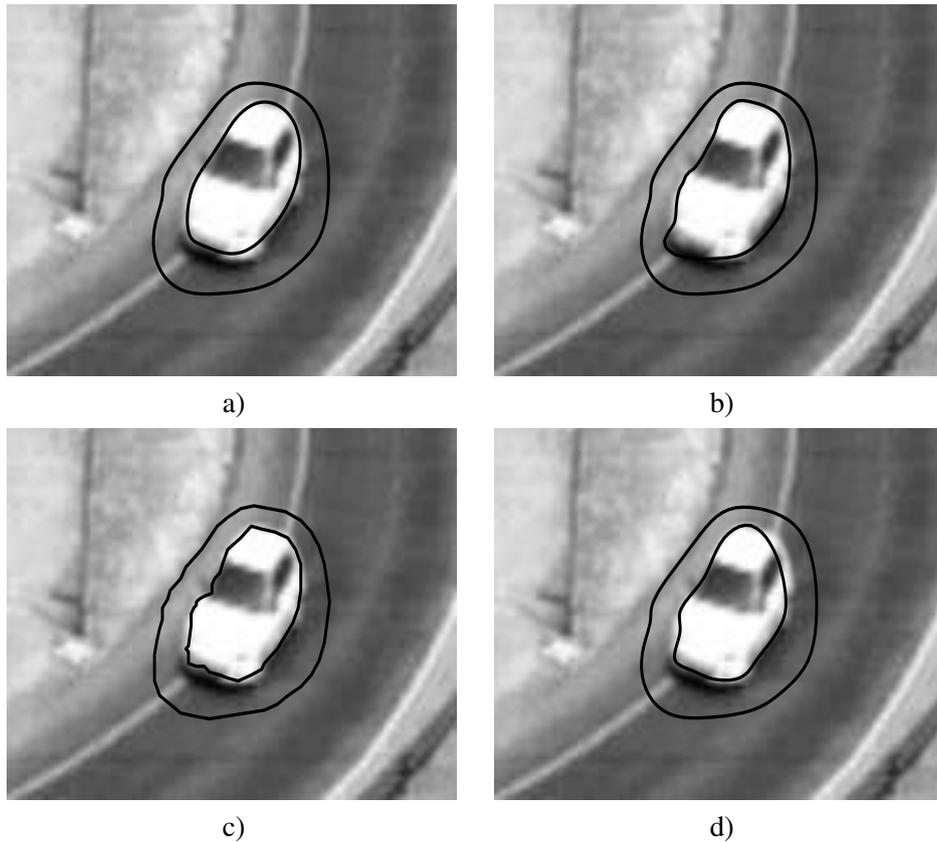


Fig. 5.15: Coche con distancia chamfer .a) Kass-ch, b) SOR-ch, c) snake-ch y d) bsnake-ch.

Imagen	Kass-ch	SOR-ch	snake-ch	bsnake-ch
Coche	0.49	0.23	6.22	30.78

Tab. 5.12: Tiempos para el coche con distancia chamfer.

En este caso el uso de  $-|G_\sigma * \nabla I|$  conduce a contornos finales erróneos mientras que la distancia chamfer hace que los contornos finales coincidan prácticamente con los del objeto.

### 5.2.3. Comentarios.

Comparando las figuras de los contornos finales obtenidos con distancia chamfer y con  $-|G_\sigma * \nabla I|$  se observa que en general el uso de la distancia chamfer como energía externa mejora los resultados y además se reduce se reduce en un factor 10

el tiempo para los métodos numéricos y a la mitad en los algoritmos voraces.

Finalmente comentar que quizá el mejor método a la vista de los resultados sea el de sobrerrelajación sucesiva (SOR) ya que los contornos finales suelen ser precisos y el tiempo en obtenerlos usando distancia chamfer es inferior a un segundo en todas las pruebas realizadas.



## 6. CONCLUSIONES

En este trabajo, partiendo de que la detección de bordes se realiza mediante la derivación numérica, se han presentado los problemas inversos y mal planteados. La derivación numérica se puede formular como un problema inverso y se puede demostrar que es un problema mal planteado. Mediante la teoría de la regularización de Tikhonov se puede restringir el espacio de soluciones al de las funciones que, por una parte se ajustan a los datos y por otra, cumplen determinadas restricciones de suavidad. Escogiendo adecuadamente el estabilizador de Tikhonov se llega a la ecuación del snake tal y como fue expuesta por primera vez aunque siguiendo una explicación menos formal.

Se han presentado dos posibles representaciones para el snake y se ha mostrado cómo adaptar la ecuación a cada representación.

Se han presentado posibles elecciones para la energía externa (la energía interna viene fijada por el estabilizador de Tikhonov): la distancia chamfer y el módulo de la convolución de una máscara gaussiana con el gradiente de la imagen todo ello con signo contrario. El objetivo de estas dos energías es que el snake sea atraído por los bordes presentes en la imagen. Finalmente se han presentado tres métodos para la minimización de la energía: dos numéricos (Kass y SOR) y uno algorítmico (algoritmos voraces).

Se han realizado pruebas sobre seis imágenes con todas representaciones y todos los métodos de minimización resultando de todo ello:

- que el uso de distancia chamfer además de reducir el tiempo de minimización considerablemente conduce a resultados más precisos en el contorno final.
- que el mejor método en cuanto a tiempos es el de sobrerrelajación sucesiva usando distancia chamfer.
- que el método que en general conduce a contornos finales más ajustados a los objetos es de minimización voraz sobre un B-Spline pero es el más costoso.

Como trabajo futuro se pretende aplicar los contornos activos a secuencias de imágenes ya sea para segmentación del movimiento como para la estimación del mismo.



## Apéndice A

### CÁLCULO DE VARIACIONES.

En este apéndice se mostrará como se obtiene la ecuación de Euler-Lagrange para un funcional del tipo:

$$\mathcal{I}(v) = \int_{s_a}^{s_b} F(s, v, v', v'') ds \quad (\text{A.1})$$

Supondremos que la función  $F$  está definida y que es dos veces derivable con continuidad respecto a  $v''$  para todos los valores de este argumento, y con respecto a  $s, v$  y  $v'$  en algún dominio  $B$  del plano.

Sea  $M$  un conjunto de curvas  $\mathcal{C}$ , entre éstas debemos hallar aquella función para la cual la integral  $\mathcal{I}(v)$  tiene el menor valor.

En el cálculo de variaciones las funciones del conjunto  $M$  se suelen denominar **admisibles**. El conjunto de funciones admisibles está determinado por las dos condiciones siguientes:

1.  $v(s)$  es continuamente diferenciable sobre el intervalo  $[s_a, s_b]$ .
2. En los extremos del segmento,  $v(s)$  toma valores prefijados:  $v(s_a) = v_1, \quad v(s_b) = v_2$

Por lo demás, la función  $v(s)$  puede ser completamente arbitraria.

Consideremos la familia de funciones dependientes de un parámetro numérico  $\alpha$ :

$$\widehat{v}(s) = v(s) + \alpha \xi(s)$$

Para que  $\widehat{v}(s)$  sea una función admisible para un  $\alpha$  arbitrario, debemos suponer que  $\xi(s)$  es continuamente diferenciable y en los extremos del intervalo  $[s_a, s_b]$  se cumple.

$$\begin{aligned} \xi(s_a) &= 0 & \xi(s_b) &= 0 \\ \xi'(s_a) &= 0 & \xi'(s_b) &= 0 \end{aligned}$$

Si realizamos la sustitución de  $v(s)$  por  $\widehat{v}(s)$  en la ecuación A.1, se tendrá que  $\mathcal{I}(\widehat{v})$  será una función del parámetro  $\alpha$ :

$$\mathcal{I}(\widehat{v}) = \int_{s_a}^{s_b} F(s, v + \alpha\xi, v' + \alpha\xi', v'' + \alpha\xi'') ds = \Phi(\alpha) \quad (\text{A.2})$$

Puesto que  $v(s)$  da un mínimo para el valor de la integral, la función  $\Phi(\alpha)$  debe tener un mínimo para  $\alpha = 0$ , por lo tanto:

$$\Phi'(0) = \int_{s_a}^{s_b} [F_y(s, v, v', v'')\xi + F_{v'}(s, v, v', v'')\xi' + F_{v''}(s, v, v', v'')\xi''] ds = 0 \quad (\text{A.3})$$

Esta última ecuación se debe verificar para toda función continuamente diferenciable  $\xi(s)$  que se anule en los extremos del segmento  $[s_a, s_b]$ . Para obtener el resultado que se deduce de esto, es conveniente transformar el segundo miembro mediante integración por partes:

$$\begin{aligned} \int_{s_a}^{s_b} F_{v'}\xi' ds &= F_{v'}\xi|_{s_a}^{s_b} - \int_{s_a}^{s_b} \xi \frac{d}{ds} F_{v'} ds = \int_{s_a}^{s_b} \xi \frac{d}{ds} F_{v'} ds \\ \int_{s_a}^{s_b} F_{v''}\xi'' ds &= F_{v''}\xi'|_{s_a}^{s_b} - \int_{s_a}^{s_b} \xi' \frac{d}{ds} F_{v''} ds = \\ &= F_{v''}\xi'|_{s_a}^{s_b} - \frac{d}{ds} F_{v''}\xi|_{s_a}^{s_b} + \int_{s_a}^{s_b} \xi \frac{d^2}{ds^2} F_{v''} ds = \int_{s_a}^{s_b} \xi \frac{d^2}{ds^2} F_{v''} ds \end{aligned}$$

Donde se ha hecho uso de las condiciones impuestas a  $\xi$ . Por tanto se puede reescribir la ecuación A.2:

$$\Phi'(0) = \int_{s_a}^{s_b} (F_v - \frac{d}{ds} F_{v'} + \frac{d^2}{ds^2} F_{v''}) \xi ds = 0 \quad (\text{A.4})$$

Si para una función arbitraria  $\xi(s)$  la integral  $\int_{s_a}^{s_b} f(s)\xi(s)ds = 0$  se puede deducir que  $f(s) = 0$  si se cumplen las condiciones siguientes:

1. La función  $f(s)$  es continua en el intervalo  $[s_a, s_B]$ .
2. La función  $\xi(s)$  es continuamente diferenciable en el intervalo  $[s_a, s_B]$  y cumplen las condiciones de contorno citadas anteriormente.

Puesto que estas condiciones se cumplen para nuestro problema podemos deducir que:

$$F_v - \frac{d}{ds} F_{v'} + \frac{d^2}{ds^2} F_{v''} = 0 \quad (\text{A.5})$$

Esta ecuación se denomina ecuación de Euler-Lagrange y finalmente podemos enunciar la siguiente conclusión:

*Si una función  $v(s)$  minimiza la integral  $\mathcal{I}$ , entonces debe verificar la ecuación diferencial de Euler-Lagrange [A.5](#)*



## Apéndice B

### DISCRETIZACIÓN DE LA ECUACIÓN DEL SNAKE.

En este apéndice se va a discretizar por el método de las diferencias finitas [5] la ecuación diferencial obtenida en el apéndice anterior. La idea que subyace en el método de las diferencias finitas para la solución de una ecuación diferencial dada es aproximar las derivadas que aparecen en la ecuación por un conjunto seleccionado de valores llamados **nodos**. El modo más usado para obtener estas aproximaciones es mediante el uso de las series de Taylor.

De este modo la primera derivada de una función  $v(s)$  en el punto  $s_i$  se puede aproximar como:

$$D v(s_i) = \frac{v(s_{i+1}) - v(s_i)}{h} + O(h) \quad (\text{B.1})$$

y la segunda derivada se puede aproximar como:

$$D^2 v(s_i) = \frac{v(s_{i+1}) - 2v(s_i) + v(s_{i-1}))}{h^2} + O(h^2) \quad (\text{B.2})$$

Si se necesita aproximar las derivadas del producto de dos funciones que dependen de  $s$  por ejemplo  $\alpha(s)v(s)$  las ecuaciones anteriores B.1 y B.2 se modifican del siguiente modo:

$$D (v(s_i)\alpha(s_i)) = \frac{v(s_{i+1})\alpha(s_{i+1}) - v(s_i)\alpha(s_i)}{h} + O(h) \quad (\text{B.3})$$

y la segunda derivada se puede aproximar como:

$$D^2 (v(s_i)\alpha(s_i)) = \frac{v(s_{i+1})\alpha(s_{i+1}) - 2v(s_i)\alpha(s_i) + v(s_{i-1})\alpha(s_{i-1}))}{h^2} + O(h^2) \quad (\text{B.4})$$

Si suponemos que  $s_k - s_{k+1} = 1 \forall k$  entonces  $h = 1$ , y liberando un poco la notación para poner  $v_i = v(s_i)$  y  $\alpha_i = \alpha(s_i)$  las ecuaciones B.3 y B.4 quedan:

$$D (v_i\alpha_i) = v_{i+1}\alpha_{i+1} - v_i\alpha_i \quad (\text{B.5})$$

$$D^2 (v_i \alpha_i) = v_{i+1} \alpha_{i+1} - 2v_i \alpha_i + v_{i-1} \alpha_{i-1} \quad (\text{B.6})$$

Con todo lo anterior ya se puede comenzar a desarrollar el tema central de este apéndice.

La ecuación diferencial de la que se parte es:

$$\frac{d}{dv} E_{ext}(v) - \frac{d}{ds} (\alpha v_s) + \frac{d^2}{ds^2} (\beta v_{ss}) = 0$$

discretizando las derivadas de  $v$  respecto de  $s$  se obtiene:

$$\frac{d}{dv} E_{ext}(v) - \frac{d}{ds} (\alpha(v_i - v_{i-1})) + \frac{d^2}{ds^2} (\beta(v_{i-1} - 2v_i + v_{i+1})) = 0 \quad (\text{B.7})$$

El segundo término de esta ecuación es:

$$\begin{aligned} \frac{d}{ds} (\alpha(v_i - v_{i-1})) &= \alpha_{i+1} v_{i+1} - \alpha_i v_i - (\alpha_{i+1} v_i + \alpha_i v_{i-1}) = \\ &= v_{i+1} \alpha_{i+1} + v_i (\alpha_{i+1} - \alpha_i) - v_{i-1} \alpha_i \end{aligned} \quad (\text{B.8})$$

El tercer término es:

$$\begin{aligned} \frac{d^2}{ds^2} (\beta(v_{i-1} - 2v_i + v_{i+1})) &= \beta_{i-1} v_{i-2} - 2\beta_i v_{i-1} + \beta_{i+1} v_i - \\ &\quad - 2(\beta_{i-1} v_{i-1} - 2\beta_i v_i + \beta_{i+1} v_{i+1}) + \beta_{i+1} v_i - \\ &\quad - 2(\beta_i v_{i+1} + \beta_{i+1} v_{i+2}) = \\ &= v_{i-2} \beta_{i-1} - 2v_{i-1} (\beta_i + \beta_{i-1}) + v_i (\beta_{i+1} + 4\beta_i + \beta_{i-1}) - \\ &\quad - 2v_{i+1} (\beta_i + \beta_{i+1}) + v_{i+2} \beta_{i+1} \end{aligned} \quad (\text{B.9})$$

Introduciendo las ecuaciones B.8 y B.9 en B.7 y si se define  $f_v(i) = \frac{d}{ds} E_{ext}(v)$  se obtiene

$$\begin{aligned} -f_v(i) &= v_{i-2} \beta_{i-1} + v_{i-1} (-2\beta_i - 2\beta_{i-1} - \alpha_i) + \\ &\quad + v_i (\beta_{i+1} + 4\beta_i + \beta_{i-1} + \alpha_{i+1} + \alpha_i) + \\ &\quad + v_{i+1} (-2\beta_{i+1} - 2\beta_i - \alpha_{i+1}) + v_{i+2} \beta_{i+1} \end{aligned} \quad (\text{B.10})$$

Teniendo en cuenta que el contorno es cerrado se tiene que

$$v_{1-1} = v_n \quad v_{1-2} = v_{n-1} \quad v_{n+1} = v_1 \quad v_{n+2} = v_2$$

Por tanto para cada valor de  $i$  se puede escribir:

$$\boxed{i=1} \quad -f_v(1) = v_{n-1}\beta_n + v_n(-\alpha_1 - 2\beta_n - 2\beta_1) + \\ + v_1(\alpha_1 + \alpha_2 + \beta_n + 4\beta_1 + \beta_2) + \\ + v_2(-\alpha_2 - 2\beta_1 - 2\beta_2) + v_3\beta_2$$

$$\boxed{i=2} \quad -f_v(2) = v_n\beta_1 + v_1(-\alpha_2 - 2\beta_1 - 2\beta_2) + \\ + v_2(\alpha_2 + \alpha_3 + \beta_1 + 4\beta_2 + \beta_3) + \\ + v_3(-\alpha_3 - 2\beta_2 - 2\beta_3) + v_4\beta_3$$

⋮

$$\boxed{i=n} \quad -f_v(n) = v_{n-2}\beta_{n-1} + v_{n-1}(-\alpha_n - 2\beta_{n-1} - 2\beta_n) + \\ + v_n(\alpha_n + \alpha_1 + \beta_{n-1} + 4\beta_n + \beta_1) + \\ + v_1(-\alpha_1 - 2\beta_n - 2\beta_1) + v_2\beta_1$$

Si se define:

$$a_i = \beta_{i-1} \\ b_i = -\alpha_i - 2\beta_{i-1} - 2\beta_i \\ c_i = \alpha_i + \alpha_{i+1} + \beta_{i-1} + 4\beta_i + \beta_{i+1}$$

podemos escribir las ecuaciones anteriores en forma matricial:



den de la posición espacial, es decir, son constantes, por tanto:

$$\begin{aligned} a_i &= \beta = a \\ b_i &= -\alpha - 2\beta - 2\beta = -\alpha - 4\beta = b \\ c_i &= \alpha + \alpha + \beta + 4\beta + \beta = 2\alpha + 6\beta = c \end{aligned}$$

con esta simplificación el sistema de ecuaciones es:

$$\begin{bmatrix} c & b & a & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a & b \\ b & c & b & a & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & a \\ a & b & c & b & a & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & a & b & c & b & a & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & a & b & c & b \\ b & a & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & a & b & c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = \begin{bmatrix} -f_v(1) \\ -f_v(2) \\ -f_v(3) \\ -f_v(4) \\ -f_v(5) \\ -f_v(6) \\ \vdots \\ -f_v(n-1) \\ -f_v(n) \end{bmatrix}$$

Esta es la ecuación que se utilizará en el presente trabajo.



## Bibliografía

- [1] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–866, September 1990.
- [2] M. Bertero. Regularization methods for linear inverse problems. Number 1225 in *Lecture Notes in Mathematics*, pages 52–112. Springer, Berlin, 1986.
- [3] Mario Bertero, Tomaso Poggio, and Vincent Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, August 1988.
- [4] Gunilla Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.
- [5] J. F. Botha and G. F. Pinder. *Fundamental Concepts in the Numerical Solution of Differential Equations*. John Wiley and Sons, New York, first edition, 1983.
- [6] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1), March 1995.
- [7] D. de Boor. *A Practical Guide to Splines*. Springer, 1978.
- [8] J. Gutiérrez, J. Albert, F.J. Ferri, and I. Llorens. Aplicación de los b-snakes a la detección de contornos en imágenes médicas. In *Actas del XV Congreso de la SEIB*, pages 84–97, Valencia, Noviembre 1997.
- [9] J. Gutiérrez, F.J. Ferri, and J. Albert. B-snakes using electrostatic potential. In *Proceedings of the VIII SNRFAI*, volume 2, pages 115–116, Bilbao, May 1999.
- [10] J. Gutiérrez, I. Llorens, F.J. Ferri, and J. Albert. Minimización de b-snakes usando algoritmos voraces y distancia chamfer. In *Actas de la VII Conferencia de la AEPIA*, pages 207–215, Torremolinos (Málaga), Noviembre 1997.
- [11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contours models. *Int. Journal Computer Vision*, 1(4):321–331, 1988.
- [12] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319, September 1985.

- [13] B. Dayanand Reddy. Functional analysis and boundary-value problems: an introductory treatment. Number 30 in Pitman Monographs and Surveys in Pure and Applied Mathematics. Longman Scientific and Technical, Essex (England), first edition, 1986.
- [14] Gilbert Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, Massachusetts, first edition, 1986.
- [15] Angus E. Taylor and David C. Lay. *Introduction to Functional Analysis*. Krieger Publishing Company, Malabar, Florida, second edition, 1980.
- [16] Demetri Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, July 1986.
- [17] Vicent Torre and Tomaso Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):147–163, March 1986.
- [18] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *GVGIP Image Understanding*, 55(1):14–26, January 1992.