

# Universitat Politècnica de Catalunya

Departament de Llenguatges i Sistemes Informàtics

Trabajo de investigación

## Contornos Activos, Mallas Activas y Cubos Activos

Realizado por:  
Máximo Gonzalo Mero Barcia

Dirigido por:  
Dr. Antoni Susin y Dra. Isabel Navazo

Tutor de doctorado:  
Dr. Pere Brunet

junio de 2001



Survey of Active Contours, Active Nets and Active Cube.

Authors: M. Mero, I. Navazo, A. Susin.

Abstract:

Deformable models give a method for computing the alteration of an object's shape due to a set of externally applied forces. Deformations may be elastic or inelastic depending on whether the original shape is restored when external forces are removed. Most deformable models in Computer Graphics result from nodal approaches, in the sense that they approximate deformations by the displacements of elementary nodes inside a flexible body. Some of them derive from the elasticity theory. Differential equations of motions are discretized in space and then integrated over time by resolving a matrix equation at each time step. This work describes three branch of the deformable models called Active Contours, Active Nets and Active Cube. These models are 2D curves, 2D nets and 3D cubes of nodes respectively controlled by physical models.



SURVEY:  
Contornos Activos, Mallas Activas y Cubos Activos

M. Mero

*Dpto. de Matemáticas - FACYT  
Universidad de Carabobo. Venezuela*

I. Navazo

*Dpto. Lenguaje y Sistemas Informáticos- LSI  
Universidad Politécnica de Cataluña-UPC. España*

A. Susin

*Dpto. de Matemáticas Aplicadas I  
Universidad Politécnica de Cataluña-UPC. España*

14 de junio de 2001

# Índice General

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Modelos Deformables</b>	<b>2</b>
<b>3</b>	<b>Contornos Activos</b>	<b>4</b>
3.1	Modelo de Contornos Activos . . . . .	5
3.2	Ventajas y Desventajas . . . . .	6
3.3	Estimación de curvatura . . . . .	8
3.4	Algoritmo Greedy . . . . .	13
3.5	Resultados de los Contornos Activos . . . . .	17
<b>4</b>	<b>Mallas Activas 2D</b>	<b>19</b>
4.1	Energía interna . . . . .	19
4.2	Energía Externa . . . . .	22
4.3	Minimización de la Energía . . . . .	25
4.3.1	Formulación Estocástica . . . . .	26
4.3.2	Mejoras deducidas del ICM . . . . .	27
4.4	Separación de una malla activa 2D . . . . .	28
4.5	Resultados Experimentales . . . . .	30
<b>5</b>	<b>Cubos Activos</b>	<b>31</b>
5.1	Resultados Experimentales . . . . .	32
5.2	Cubos Activos en simulación quirúrgica . . . . .	33
<b>6</b>	<b>Conclusiones</b>	<b>36</b>
<b>7</b>	<b>Apéndice</b>	<b>37</b>
7.1	Conceptos elementales de probabilidades y estadística . . . . .	37
7.2	Preliminares a los Campos Aleatorios de Markov y Distribución de Gibbs . . . . .	41
7.3	Campos Aleatorios de Markov (CAM) . . . . .	43
7.4	Distribución de Gibbs . . . . .	44
7.5	Simulated Annealing . . . . .	44
7.6	Programación Dinámica . . . . .	45

22	Vecinos en una rejilla regular $S$ . (a) Sistema de primer orden o de 4-vecinos. (b) Sistema de segundo orden o de 8-vecinos. (c) Los números indican los estados vecinos exteriores de orden $n = 1, \dots, 5$ . . . . .	42
23	Cliques en una rejilla regular . . . . .	43

## Índice de Tablas

1	Comparación de estimados del cuadrado de la curvatura usando diferentes métodos	11
2	Comparación del tiempo de ejecución, número de iteraciones, y número de discontinuidades de segundo orden realizados con los métodos Alg. Greedy (AG), Prog. Dinámica (PD) y Cálculo Variacional (CV) . . . . .	23

# Índice de Figuras

1	Parametrización de diferentes familias de Snake. La superior izquierda: contorno activo, superior derecha: superficie activa, inferior izquierda: malla activa, inferior derecha: cubos activos . . . . .	4
2	Contorno indicando las características principales del objeto en vez de la textura de aristas sobre la superficie del mismo . . . . .	8
3	Posición de los puntos para la tabla 1 . . . . .	10
4	Diferencia en dirección de dos vectores . . . . .	12
5	Estimado de curvaturas al fijar tres puntos en un círculo: (a) el ángulo entre vectores es grande, por lo tanto la aproximación en este caso es bastante buena. (b) ángulo muy pequeño, no es muy bueno como aproximación de curvatura a través de los tres puntos. (c) Aún cuando la distancia entre punto no es la misma y están dispuestos en diferentes posiciones con aparente curvatura distinta, no lo es así en estos casos, donde varias ternas de puntos sobre el círculo tienen la misma curvatura . . . . .	13
6	La función de energía es calculada en cada $v_i$ y en cada uno de sus 8 vecinos. Los puntos anteriores y posteriores a dicho punto del contorno, son usados en el cálculo de la restricción de continuidad. La posición que tiene el mínimo valor de energía es elegida como una nueva posición de $v_i$ . . . . .	16
7	Cuadrado:(a)Contorno original, (b)Método de Kass, (c)Algoritmo de programación Dinámica, (d)Algoritmo Greedy. . . . .	19
8	Caja:(a)Contorno original, (b)Método de Kass, (c) Algoritmo de programación Dinámica, (d)Algoritmo Greedy. . . . .	20
9	Botella:(a)Contorno original, (b)Método de Kass, (c)Algoritmo de programación Dinámica, (d)Algoritmo Greedy. . . . .	21
10	Taza:(a)Contorno original, (b)Método de Kass, (c)Algoritmo de programación Dinámica, (d)Algoritmo Greedy. . . . .	22
11	Segmentación de un objeto en forma de anillo por un contorno activo. . . . .	23
12	Convergencia de un contorno activo hacia una imagen, obtenida por CT-scan, de una vértebra. . . . .	24
13	Cliques en una rejilla regular . . . . .	26
14	Corte de malla no apropiado . . . . .	30
15	Colocación de prioridades después que un arco es cortado . . . . .	30
16	Prueba del algoritmo de división de una malla. En las imágenes del medio se ve como se va efectuando la división de la malla activa, y en la última imagen se aprecia el resultado final de esta división de la malla en dos sub-mallas. . . . .	31
17	Resultado al aplicar el algoritmo de cubo activo a un CT-scan de una cabeza. . . . .	33
18	Interior de un Active net 3D. . . . .	33
19	Objeto a ser simulado usando Active Cube. . . . .	34
20	Modelado del objeto y el movimiento de tejido duro usando cubos activos. . . . .	34
21	Resultado final, para el movimiento de tejido suave, usando cubo activo. . . . .	34

# 1 Introducción

Los modelos son esquemas teóricos de un sistema o de una realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento. Por modelos deformables se entienden los esquemas teóricos de un sistema o una realidad que se modifica y su respectivo análisis. Los modelos activos deformables, familia de la clase de los modelos deformables, son modelos de un sistema controlado por características del modelo físico, definidas en términos de energías que hace que el mismo se modifique continuamente.

Los modelos activos han llamado mucho la atención en años recientes por parte de los investigadores de la comunidad de computación gráfica, visualización entre otras, así como por sus aplicaciones en áreas como, por ejemplo medicina. Además se han presentado una gran variedad de esquemas para la optimización de la energía del modelo. En este trabajo haremos una revisión de estos modelos activos centrandó nuestra atención en tres de ellos. Active Contours, Active Nets y Active Cube. Estos modelos son curvas cerradas 2D, mallas 2D y cubos 3D respectivamente, controlados por los modelos físicos.

El modelo de la snake fue propuesto inicialmente por M. Kass, A. Witkin and D. Terzopoulos [MAD88]. Luego ha sido extensamente usado en problemas de aplicaciones de análisis de imágenes y visualización, tales como extracción de contornos, segmentación de datos 2D entre otras. Consideramos la aplicación de modelos de contornos deformables como un método para extraer formas “suaves” en una región u objeto de la imagen. La filosofía del enfoque es introducir una curva elástica en la imagen, y dejar que se despliegue desde su posición inicial bajo la acción de fuerzas internas (restricciones de suavidad y fuerzas de presión) y de fuerzas externas (fuerzas de atracción de aristas y fuerzas de peso). Las fuerzas internas y externas están definidas para ajustarse a una frontera de objeto u otras características deseadas dentro de la imagen.

Existen dos enfoques básicos para segmentación y etiquetado de la imagen. En el enfoque clásico, las características son extraídas de la imagen, obteniendo una colección dispersa de datos, luego se usan los métodos de reconstrucción para interpolar estos datos para formar así una representación (y posible segmentación) de los datos originales. En los enfoques más recientes, tales como el método de los Contornos Activos (Snake), se da una estimación inicial de la curva y se usan los métodos de optimización para refinar la estimación inicial a partir de los datos de la imagen.

En el trabajo propuesto por Kass del modelo activo, se encuentra una solución usando técnicas de Cálculo Variacional. En el trabajo de Amini[AAT88] se detallan algunos de los problemas de este enfoque, incluyendo la inestabilidad numérica y la tendencia a que los puntos se agrupen en porciones de una arista del contorno. Como solución propusieron un algoritmo para el

modelo de contorno activo usando Programación Dinámica. Este enfoque es más estable y permite la inclusión de restricciones duras además de las restricciones suaves, inherentes a la formulación de la funcional de energía, sin embargo esto es lento, teniendo una complejidad del orden  $O(nm^3)$ , donde  $n$  es el número de puntos en el contorno y  $m$  es el tamaño de la vecindad en la cual un punto puede moverse en una iteración simple. En este trabajo se resume la solidez y la debilidad de estos enfoques y se muestra un algoritmo (Greedy Algorithm) el cual tiene una ejecución comparable al obtenido con Programación Dinámica y Cálculo Variacional. Este enfoque conserva las mejoras de estabilidad, flexibilidad y la inclusión de restricciones duras introducidas por el método de Programación Dinámica, pero es un orden de magnitud más rápido que dicho enfoque, siendo el orden  $O(nm)$ . Una formulación diferente, a la de los autores previos, es usada por Williams y Shah[DM92] para el término de continuidad, el cual logra que los puntos en el contorno estén uniformemente espaciados. La idea de uniformidad de distancia entre nodos hace que el cálculo de la curvatura sea más exacto, ya que el concepto de curvatura es básico para la formulación del funcional de energía del contorno.

Las mallas activas o Active Nets fue originalmente propuesto en 1989 por Y. Tsumiyama, K. Sakaue y K. Yamamoto [YKK89]. Al igual que los contornos activos, las mallas activas se deforman bajo las influencias de fuerzas internas y fuerzas externas. Las fuerzas internas representan propiedades heredadas del modelo físico tales como contracción y estabilidad de la forma. Las fuerzas externas hacen que la malla se contraiga hacia características de interés de la imagen las cuales normalmente son regiones de alta intensidad. Se pueden ver las mallas activas como una extensión del enfoque de contornos activos propuesto por Kass[MAD88], que también puede ser aplicado a los cubos activos o Active Cubes (la versión volumétrica).

Las imágenes mostradas en este documento corresponden a resultados, de las aplicaciones de los modelos deformables activos, de trabajos de investigadores en el área. De estas resaltan los obtenidos por Bro-Nielsen en el campo de la medicina.

## 2 Modelos Deformables

La familia de modelos Activos Deformables que mostraremos tiene la siguiente parametrización:

$$\left\{ \begin{array}{l} [0, 1]^m \rightarrow R^n \\ \mathbf{p} = (r, s, \dots) \rightarrow \mathbf{v}(\mathbf{p}) = (x(r, s, \dots), y(r, s, \dots), \dots) \\ \text{Condiciones de fronteras} \end{array} \right. \quad (1)$$

con deformaciones controladas por una función de energía:

$$\begin{cases} E : A \subset R^n \rightarrow R \\ \mathbf{v} \rightarrow E(\mathbf{v}) = \int_{\dots} \int E_{int}(\mathbf{v}(r, s, \dots)) + E_{ext}(\mathbf{v}(r, s, \dots)) dr ds \dots \end{cases} \quad (2)$$

donde  $A$  es el conjunto de deformaciones admisibles y  $E_{int}$  define la energía interna del modelo, que mide la energía de desviación del modelo desde los requerimientos de formas dados.  $E_{ext}$  representa las fuerzas externas actuando sobre el modelo, es decir, las fuerzas de atracción de la imagen. El estado óptimo para el modelo activo es que  $\mathbf{v} \in A$ , las cuales minimizan la energía  $E(\mathbf{v})$ .

Cada elemento de esta familia puede ser descrita por la dimensión de parametrización  $m$  y la dimensión de imagen  $n$ , junto con el conjunto de condiciones de fronteras. Para ver estas topologías, ver la figura(1).

Los contornos activos han sido propuestos como un método efectivo para detección de contorno. La base de estos modelos son curvas definidas dentro de un dominio de imagen que puede moverse bajo la influencia de fuerzas internas obtenidas de la misma curva y de fuerzas externas calculadas de los datos de la imagen. Sin embargo, el modelo tradicional de las snake, nombre con el cual se denominan los contornos activos, tiene como punto débil que el contorno inicial debe estar próximo a la frontera real, puesto que el modelo no es capaz de encontrar fronteras que están muy alejadas.

La ventaja de las Active Nets con respecto a las Active Contours [MAD88] (y los Ballons [LI84]) es la existencia de nodos internos (nodos dentro de la frontera de la Net) los cuales mejoran el contenido de información de la imagen.

Con los Active Cubes el interior de un volumen 3D está definido ya que también se tienen los nodos internos. En contraste, los modelos de superficies, que tradicionalmente han sido usados para simular operaciones médicas, no contienen ninguna información acerca de objeto bajo la superficie, como consecuencia, no es posible cortar una superficie 3D con un resultado realista. Con los Active Cubes el interior está definido y es posible eliminar nodos de superficie revelando el interior del volumen en forma realista. Un ejemplo importante en simulación quirúrgica es modelar la deformación global del tejido suave en un paciente cuando los huesos se mueven en su interior. Los Active Cubes pueden ser usados para este propósito.

Estructuras similares a Active Nets y active Cubes han sido usadas en animación en computación gráfica [D.T88] y para el modelado de estructuras físicas.

Para la deformación a lo largo del tiempo de la familia de los modelos activos, Active contours, Active nets y Active cube, se han presentado muchas técnicas referentes a la minimización de la energía. Estas se pueden dividir rigurosamente en dos clases basadas en la forma que usan métodos de análisis determinísticos o no determinísticos para calcular el estado de minimización de energía del modelo activo:

- Análisis Numérico
  - Diferencias Finitas
  - Elementos Finitos
- Fuerza Bruta
  - Programación Dinámica
  - Algoritmo Greedy
- Otros
  - Fundamentos Probabilísticos

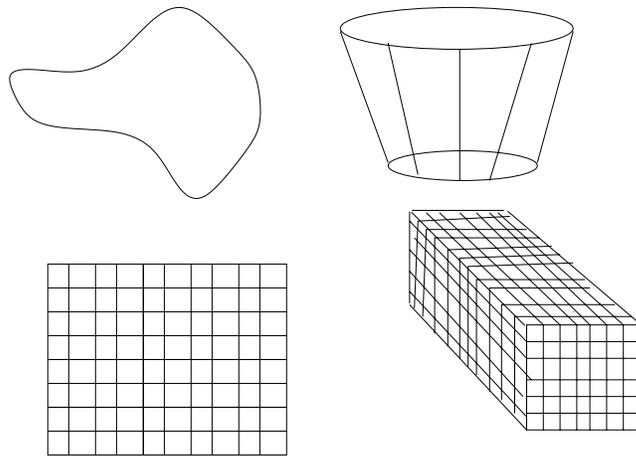


Figura 1: *Parametrización de diferentes familias de Snake. La superior izquierda: contorno activo, superior derecha: superficie activa, inferior izquierda: malla activa, inferior derecha: cubos activos*

### 3 Contornos Activos

Los modelos de contornos activos son una manera efectiva de extraer contornos combinando características de imágenes, factores de regularización y otras restricciones.

Estos modelos deformables tienen dos tipos de energías asociadas: una energía interna que caracteriza los atributos deseados del contorno, y una energía externa que ata el contorno con lo esencial de la imagen. El objetivo es la minimización de la suma de estas energías.

La filosofía del enfoque es introducir una curva elástica en la imagen, y dejar que se despliegue desde su posición inicial bajo la acción de fuerzas internas

(fuerzas elásticas, restricciones de suavidad y presión) y de fuerzas externas (fuerzas de imágenes, fuerzas de restricciones como fuerzas de atracción hacia aristas y fuerzas de peso), hacia la frontera de la imagen.

Kass, Witkin y Terzopoulos[MAD88] propusieron un modelo llamado contornos Activos o Snake, el cual tiene la ventaja de que la forma definitiva de un contorno puede estar influenciada por feedback desde un nivel avanzado del proceso. El contorno es colocado inicialmente cerca de una arista de importancia, luego las fuerzas de imagen mueven el contorno hacia la arista en la imagen. Como el algoritmo es iterativo, el término de energía puede ser ajustado en algún nivel del proceso para obtener un mínimo local que sea más útil para dicho nivel. Sin embargo, existen algunos problemas con el procedimiento de minimización empleado. Amini, Tehrani y Weymouth[AAT88] señalaron algunos de estos problemas, incluyendo inestabilidad numérica y una tendencia de agrupamientos de puntos sobre porciones de aristas fuertes. Ellos propusieron un algoritmo de Programación Dinámica [ver apéndice] para minimizar la funcional de energía que permita la inclusión de restricciones duras para así obtener un comportamiento más deseable de la Snake.

En el trabajo de Williams and M.Shah[DM92] se señalan algunos de los problemas con ambos métodos y se muestra un algoritmo, el cual es estable, flexible, permite la inclusión de restricciones duras y se ejecuta mucho más rápido que el método de Programación Dinámica. También al funcional de energía se le agrega una nueva formulación del término de continuidad y del término de curvatura. Se usa una formulación diferente para el término de continuidad de tal manera que los puntos estén uniformemente espaciados sobre el contorno, en vez de la minimización de la distancia entre puntos como en los métodos previos. También es importante una aproximación discreta de la curvatura de manera exacta y eficiente, por lo tanto, en los apartados siguientes, se examinan algunas aproximaciones de la curvatura y las mismas se comparan.

### 3.1 Modelo de Contornos Activos

En la actualidad es común usar más de un patrón, para detectar aristas o representar contornos[DE80] [J.C86]. En vez de combinar la información deducida en las diferentes escalas hacia la unificación de la "mejor" representación de la información, otro enfoque es intentar mantener la información en diferentes escalas, disponibles para que en un nivel avanzado del proceso, pueda ser usada la representación más útil. Esta fue una de las metas de Kass[MAD88]y sus colegas, cuando desarrollaron su Modelo de Contorno Activo llamado Snake. Ellos desarrollaron un spline de continuidad controlada, el cual puede ser operado por fuerzas internas de contornos, fuerzas de imagen, y fuerzas externas, las cuales pueden ser suministradas de forma interactiva por el usuario.

En su trabajo, Kass representó un contorno por un vector  $\mathbf{v}(s) = (x(s), y(s))$ , con  $s$  como parámetro de longitud de arco. El funcional de energía se escribe como:

$$\begin{aligned} E_{snake}^* &= \int_0^1 E_{snake}(\mathbf{v}(s)) ds \\ &= \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{cont}(\mathbf{v}(s)) ds \end{aligned} \quad (3)$$

$E_{int}$  representa la energía interna del contorno, debido a contracciones, doblamiento o discontinuidades,  $E_{ext}$  son las fuerzas de imagen y  $E_{con}$  son las restricciones externas. Las fuerzas de imagen pueden ser debidas a distintos motivos. Los que Kass presenta son líneas, aristas y puntos fronteras. El spline de energía interna viene dado por:

$$E_{int} = (\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2)/2 \quad (4)$$

La ecuación anterior contiene un término de primer orden, el cual tendrá valores grandes cuando exista un gap en la curva, y un término de continuidad de segundo orden, el cual será grande cuando la curva se doble rápidamente. Los valores de  $\alpha$  y  $\beta$  en un punto, determinan la extensión para la cual se le permite al contorno estirarse o doblarse en ese punto.  $\alpha$  y  $\beta$  pueden ser escogidos para controlar la influencia de las respectivas restricciones. Por ejemplo, un valor grande de  $\beta$  haría del término de continuidad de segundo orden, más grande que cualquiera de los otros términos, así, el mínimo valor de  $E^*$  se obtendrá cuando la curva sea más suave, aproximándose a un círculo para un contorno cerrado, y a una línea recta para un contorno no cerrado. Si  $\alpha$  es 0 en un punto, una discontinuidad puede ocurrir en ese punto, mientras si  $\beta$  es 0, puede aparecer una esquina, puesto que valores grandes de estos términos no serían incluidos en la suma. Hecho este planteamiento, se pueden usar técnicas de Cálculo Variacional para determinar el mínimo del funcional de energía.

Con la propuesta de Amini, usando Programación Dinámica, se permite la introducción de restricciones que no pueden ser violadas, llamadas restricciones duras, así como también restricciones de continuidad de primer y segundo orden, las cuales son inherentes a la formulación del problema. Estas últimas restricciones son conocidas como suaves.

### 3.2 Ventajas y Desventajas

A continuación se indican las ventajas y desventajas de la formulación del problema de estos dos métodos de solución.

Ventajas generales.

- Un contorno cerrado, que sea colocado alrededor de un objeto puede generar gaps, en el mapa de arista. Análogamente, si un objeto con

textura tiene aristas que la hacen aparecer como pequeños objetos, el contorno puede perfilar el objeto como un todo, dando un contorno de arista continuo para el objeto completo. Ver figura(2).

- La información en varios niveles del proceso puede ser usada para determinar valores del término de las restricciones externas y los valores  $\alpha$  y  $\beta$ . Por ejemplo, las esquinas serían permitidas en ciertos puntos y se examinará el efecto sobre el contorno examinado.

Desventajas generales.

- No se da ninguna indicación en ninguno de los métodos para determinar los valores de  $\alpha$  y  $\beta$ . También, ambos métodos aparentemente usan el mismo valor de  $\alpha$  y  $\beta$  en cada punto, y no discuten o explican con ejemplos, como cambiando estos valores, se afecta el contorno. Por lo tanto, estos valores son trascendentales, y deben ser elegidos cuidadosamente para obtener resultados satisfactorios.
- Relacionado con los puntos previos, si  $\beta$  es constante, las esquinas no estarían bien definidas. Existen también problemas si los puntos están muy apartados y una esquina está entre dichos puntos sobre un contorno.
- La primera derivada en la ecuación ( 4) es aproximada por diferencias finitas,  $|\mathbf{v}_s|^2 \approx (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$ . Esto es equivalente a minimizar la distancia entre puntos, y tiene el efecto de hacer que el contorno se encoja.
- Los puntos pueden moverse a lo largo del contorno haciendo que se agrupen puntos en segmentos del contorno donde las fuerzas de imagen son mayores. Las restricciones duras dadas en el método de Amini pueden minimizar este problema.

Ventajas del método propuesto por Kass.

- Las fuerzas pueden recorrer largas distancias a lo largo del contorno en una iteración, haciendo la convergencia más rápida.

Desventajas del método propuesto por Kass.

- Las fuerzas de imagen y las restricciones deben ser diferenciables para garantizar convergencia. De esta forma no es posible incluir restricciones duras, tales como distancia mínima entre puntos.

Ventajas del método propuesto por Amini.

- Se pueden introducir restricciones duras en el método.

- Los puntos son movidos sobre una retícula discreta, opuesto al método de Kass, el cual calcula las coordenadas de puntos como números reales, haciendo que los puntos caigan entre las coordenadas discretas.
- Este método es numéricamente estable.

Desventajas del método propuesto por Amini.

- Los requerimientos de memoria son más grandes, siendo del orden  $O(nm^2)$ , donde  $n$  es el número de puntos sobre el contorno y  $m$  es el número de posibles localizaciones para el cual un punto puede moverse en una iteración simple.
- El método es muy lento, siendo del orden  $O(nm^3)$ .

En el trabajo de Williams[DM92] se resuelven muchas de las dificultades de los métodos anteriores. Con dicho método se preservan la inclusión de restricciones duras, el uso de una retícula discreta para la posición de los puntos y la estabilidad que fueron logradas con el método de Programación Dinámica. Además, la elección de los valores para  $\alpha$ ,  $\beta$  y  $\gamma$  (un nuevo parámetro) son fácilmente determinados para balancear las fuerzas relativas de los términos en la funcional de energía. Una reformulación del término de continuidad de primer orden hace que los puntos estén uniformemente espaciados sobre el contorno, eliminando el comportamiento de "encogimiento" del contorno y haciendo que la estimación del término de segundo orden de continuidad sea más exacto.

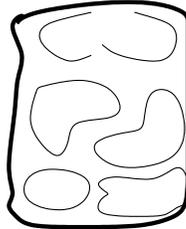


Figura 2: Contorno indicando las características principales del objeto en vez de la textura de aristas sobre la superficie del mismo

### 3.3 Estimación de curvatura

Tanto Kass como Amini aproximan las derivadas en la ecuación ( 4) por diferencias finitas. Si  $\mathbf{v} = (x_i, y_i)$  es un punto del contorno, se usan las siguientes aproximaciones:

$$\left| \frac{d\mathbf{v}_i}{ds} \right|^2 \simeq |\mathbf{v}_i - \mathbf{v}_{i-1}|^2 = (\mathbf{x}_i - \mathbf{x}_{i-1})^2 + (\mathbf{y}_i - \mathbf{y}_{i-1})^2 \quad (5)$$

y

$$\left| \frac{d^2\mathbf{v}_i}{ds^2} \right|^2 \simeq |\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}|^2 = (\mathbf{x}_{i-1} - 2\mathbf{x}_i + \mathbf{x}_{i+1})^2 + (\mathbf{y}_{i-1} - 2\mathbf{y}_i + \mathbf{y}_{i+1})^2 \quad (6)$$

Además, se consideran dos suposiciones. La primera, es que los puntos están uniformemente espaciados en intervalo unitario. Puesto que los puntos están uniformemente espaciados, entonces la ecuación ( 5) estaría dividida por  $d^2$ , donde  $d$  es la distancia entre puntos, y la ecuación ( 6) por  $d^4$ . Esto no representa problema alguno ya que los valores  $\alpha$  y  $\beta$  pueden ser elegidos para que incluyan estos factores. En este caso  $d$  tendría que estar disponible para ser usado en cualquier nivel del proceso en el cual se intenta asignar valores a  $\alpha$  y  $\beta$ .

Si los puntos no están uniformemente espaciados, el término de la primera derivada sería incorrecto por un factor  $d_i^2$ , donde  $d_i$  es la distancia entre los puntos  $i$  e  $i - 1$ . Esto hará que el término de continuidad de primer orden en la expresión de energía sea mayor para puntos que están muy alejados. Además, el término de la segunda derivada daría un estimado de curvatura grande.

La segunda suposición es que la parametrización se hace respecto de la longitud de arco. Cuando se hace esta suposición, la curvatura viene dada por  $|\mathbf{v}_{ss}|$ . Sin embargo, cuando el parámetro no es la longitud de arco, la curvatura viene dada por:

$$\kappa = \frac{|x'y'' - x''y'|}{(x'^2 + y'^2)^{3/2}} \quad (7)$$

para  $t$  el parámetro y donde  $x' = dx/dt$  y  $x'' = d^2x/dt^2$   $y' = dy/dt$  y  $y'' = d^2y/dt^2$ .

No es claro que medida de curvatura es la que mejor refleja la situación geométrica dibujada por el contorno. La definición matemática de curvatura es  $d\theta/ds$ , donde  $\theta$  es el ángulo entre el eje positivo  $x$  y el vector tangente a la curva. Esta es una medida independiente del sistema de coordenadas, así, cuando cualquier recta es sustituida por el eje  $x$ , esta medida es invariante por rotación. Esta es una de las características deseables para ajustar modelos. Otra característica deseable, la cual no viene dada por la curvatura, es la invariabilidad por escalamiento. Un círculo con radio  $r$  tiene curvatura  $1/r$  en cada punto. De está forma, cuando el radio se duplica, la curvatura se divide por la mitad.

A continuación se presentan 5 medidas posibles de curvatura en contornos discretos y se discuten las características de cada una. Para demostrar la diferencia en los resultados obtenidos por estas aproximaciones, los distintos métodos se aplicaron en las situaciones dadas en la figura(3.a) y los resultados se muestran en la tabla [1]. En cada uno de los casos de la figura(3),  $v_{i-1}$  es el punto  $a$ ,  $v_i$  es el punto  $b$ , mientras  $v_{i+1}$ , el tercer punto en el estimado de la curvatura, puede ser cualquiera de los puntos  $c_1 \dots c_5$ . La primera sección de la tabla es la situación en la figura(3.a), donde  $a$  y  $b$  son líneas horizontales o verticales. La sección inferior es el caso cuando  $a$  y  $b$  yacen sobre una recta diagonal. Cuando el ángulo externo es  $0$ ,  $\pi/2$  o  $\pi$ , la distancia de  $b$  a sus dos vecinos es la misma, siendo  $1$  para el caso horizontal y  $\sqrt{2}$  para el caso diagonal. Cuando el ángulo es  $\pi/4$  o  $3\pi/4$  las dos distancias no son iguales, siendo  $1$  y  $\sqrt{2}$ .

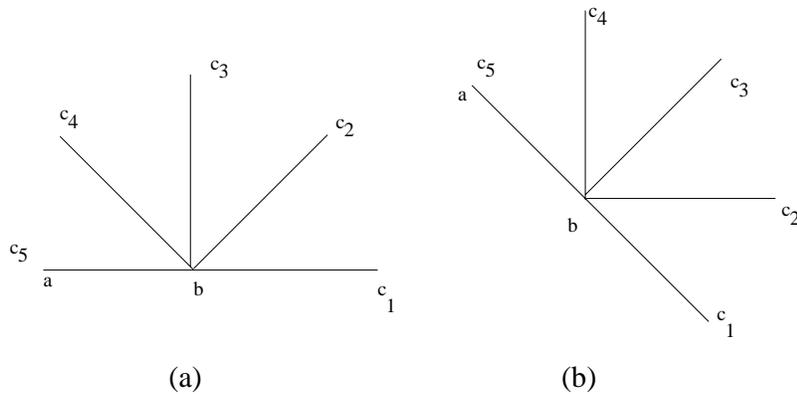


Figura 3: Posición de los puntos para la tabla 1

Es necesario estimar los valores de los diferenciales en la discusión que sigue. La convención usual es la diferencia hacia “atrás” para este estimado.

Esto es,  $dx$  en el punto  $v_i$  se aproxima por  $(x_i - x_{i-1})$  y se denota por  $\Delta x_i$ . Ocasionalmente, cuando las diferencias hacia atrás pueden variar sustancialmente de las diferencias hacia adelante,  $(x_{i+1} - x_i)$ , se usará la diferencia centrada  $(x_{i-1} - x_{i+1})$ . Cuando esto se hace, se indica. Una notación similar para todas las estimaciones de diferencias finitas son utilizadas.

La primera posibilidad para aproximar curvas es aplicar la definición de curvatura directamente. Si una aproximación discreta de  $d\theta/ds$  es calculada para puntos uniformemente espaciados, introducimos los vectores

$\vec{u}_i = (x_i - x_{i-1}, y_i - y_{i-1})$  y  $\vec{u}_{i+1} = (x_{i+1} - x_i, y_{i+1} - y_i)$  y la fórmula para  $\Delta\theta$  vendrá dada por

c	$(d\theta/ds)^2$	$\kappa$	$ \mathbf{v}_{ss} ^2$	$ \vec{\mathbf{u}}_i - \vec{\mathbf{u}}_{i+1} ^2$	$ \frac{\vec{\mathbf{u}}_i}{ \vec{\mathbf{u}}_i } - \frac{\vec{\mathbf{u}}_{i+1}}{ \vec{\mathbf{u}}_{i+1} } ^2$
		Caso Horizontal			
1	0.0	0.0	0.0	0.0	0.0
2	0.42	0.512	0.40	1.0	0.59
3	2.47	8.0	2.0	2.0	2.0
4	3.8	64.0	2.34	5.0	3.41
5	9.87	$\infty$	2.0	8.0	4.0
		Caso Diagonal			
1	0.0	0.0	0.0	0.0	0.0
2	0.42	0.512	0.40	1.0	0.59
3	1.23	4.0	1.0	4.0	2.0
4	3.8	64.0	2.34	5.0	3.41
5	4.93	$\infty$	2.0	8.0	4.0

Tabla 1: Comparación de estimados del cuadrado de la curvatura usando diferentes métodos

$$\begin{aligned} \Delta\theta &= \cos^{-1}\left(\frac{\vec{\mathbf{u}}_i \cdot \vec{\mathbf{u}}_{i+1}}{|\vec{\mathbf{u}}_i| |\vec{\mathbf{u}}_{i+1}|}\right) \\ &= \cos^{-1}\left(\frac{(x_i - x_{i-1})(x_{i+1} - x_i) + (y_i - y_{i-1})(y_{i+1} - y_i)}{\sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2][(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]}}\right) \end{aligned}$$

Dado un polígono cerrado y una dirección,  $\Delta\theta \approx d\theta$  es el ángulo externo medido en sentido contrario a las agujas del reloj. La expresión  $(\Delta s_{i+1} + \Delta s_i)/2 = (|\vec{\mathbf{u}}_{i+1}| + |\vec{\mathbf{u}}_i|)/2$  promedia la distancia del punto  $i$  a sus dos vecinos y esto da el mejor estimado de  $ds$ . El valor más grande de  $\Delta\theta$  es  $\pi$ , por lo tanto  $\Delta\theta/\Delta s$  caen en el intervalo  $[0, \pi]$ . Esto no es cierto para curvas continuas, donde las curvas con un doblamiento rápido pueden tener grandes valores de curvaturas. La columna 1 de la tabla [1] da valores para  $(d\theta/ds)^2$ .

Evaluando la expresión para curvatura en la ecuación ( 7) se darían resultados idénticos a los de  $(d\theta/ds)^2$  para curvas continuas. sin embargo, esto no es el caso para curvas discretas. Cuando el ángulo entre  $\vec{\mathbf{u}}_i$  y  $\vec{\mathbf{u}}_{i+1}$  es grande,  $\Delta x$  tendría un valor próximo a  $-\Delta x_{i+1}$  y  $\Delta y_i$  próximo a  $-\Delta y_{i+1}$ . De esta forma, cuando la diferencia centrada es usada para estimar  $dx$  y  $dy$ , estos valores serían muy pequeños, dando un valor de curvatura que es no acotado, como lo es para el caso de curvas continuas. En la columna 2 de la tabla [1], la aproximación discreta a la ecuación ( 7), es similar a los otros estimados para ángulos pequeños, pero como  $(\Delta x_{i+1} + \Delta x_i)/2$  y  $(\Delta y_{i+1} + \Delta y_i)/2$  aumenta poco, los estimados para  $\kappa^2$  serían muy grandes.

Convirtiendo el parámetro a longitud de arco y luego calculando la segunda derivada, es teóricamente igual a las dos medidas anteriores para curvas continuas. La aproximación discreta viene dada por:

$$\mathbf{v}_{ss}(s) = \frac{1}{\Delta s} \sqrt{\left(\frac{\Delta x_i}{\Delta s_i} - \frac{\Delta x_{i+1}}{\Delta s_{i+1}}\right)^2 + \left(\frac{\Delta y_i}{\Delta s_i} - \frac{\Delta y_{i+1}}{\Delta s_{i+1}}\right)^2} \quad (8)$$

donde  $\Delta s$  es  $(\Delta s_{i+1} + \Delta s_i)/2$ . La tercera columna de la tabla [1] da el cuadrado del estimado discreto de la segunda derivada. Note en el caso diagonal en la columna 3 que la curvatura para  $c_4$  es mayor que en  $c_5$ , aunque el camino  $a - b - c_5$  realmente duplica el recorrido sobre sí mismo, y tendría, intuitivamente, mayor curvatura.

Otra medida de curvatura, la cual tiene la ventaja de ser computacionalmente eficiente, viene dada por la expresión en la ecuación (6). Si  $\vec{\mathbf{u}}_1$  y  $\vec{\mathbf{u}}_2$  son los vectores definidos anteriormente, la curvatura es equivalente a  $|\vec{\mathbf{u}}_1 - \vec{\mathbf{u}}_2|^2$ . Esto refleja no solo las diferencias entre las direcciones de los dos vectores sino también la diferencias en longitud. Así, si los tres puntos no están igualmente espaciados, la curvatura será mayor. Por ejemplo  $a, b$  y  $c$  pueden yacer sobre una línea recta y la curvatura no sería cero. La cuarta columna en la tabla es el cuadrado de la curvatura estimada usando la ecuación (6). Nótese que en la columna 4,  $c_4$  tiene el valor más grande para el caso horizontal como lo es en la columna 3 para el caso diagonal.

Normalizando los dos vectores antes de tomar la diferencia hace que la medida dependa solamente de la dirección relativa. De esta forma, la curvatura estará acotada, con valores en el intervalo  $[0,2]$ . La longitud de  $\vec{\mathbf{u}}_{i+1}/|\vec{\mathbf{u}}_{i+1}| - \vec{\mathbf{u}}_i/|\vec{\mathbf{u}}_i|$  viene dada por  $2 \sin(\theta/2)$ , donde  $\theta \in [0, \pi]$  es la diferencia en la dirección de los dos vectores como se muestra en la figura(4). La columna 5 de la tabla da los valores obtenidos por esta fórmula.

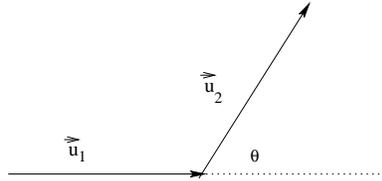


Figura 4: Diferencia en dirección de dos vectores

Es interesante ver que las últimas 3 medidas están estrechamente relacionadas. Multiplicando la aproximación discreta de  $|\mathbf{v}_{ss}|$  por  $\Delta s$  se obtienen las diferencias de los vectores normalizados. Cuando los puntos están igualmente espaciados, multiplicando de nuevo por  $\Delta s$  se obtiene la expresión en la ecuación (6).

Existe un sexto método de aproximación de curvatura en un punto, el cual ajusta un círculo a través del punto y sus dos vecinos. El radio del círculo dará un buen estimado del radio de curvatura si el círculo es una buena aproximación de la curva a través de los tres puntos. Sin embargo,

esto solo da un estimado razonable cuando el ángulo entre los dos vectores es grande y cuando los puntos están igualmente espaciados, ver figura(5.a). Cuando el ángulo entre los dos vectores es pequeño, el círculo no da una buena aproximación a la curva a través de los tres puntos y el estimado de curvatura sería muy pequeño, figura(5.b). Si los puntos no están igualmente espaciados, diferentes situaciones, las cuales parecen que no tengan la misma curvatura, darían un círculo con el mismo radio para cada una de ellas, figura(5.c). Por ejemplo, el conjunto de puntos  $\{a, b, c\}$ ,  $\{a, b, d\}$  y  $\{a, b, e\}$  tendrían un círculo del mismo radio fijado a través de ellos, aunque la curvatura de la curva a través de los diferentes conjuntos no parezcan el mismo. Así, este método no parece tener una aplicación general para ser considerada aquí.

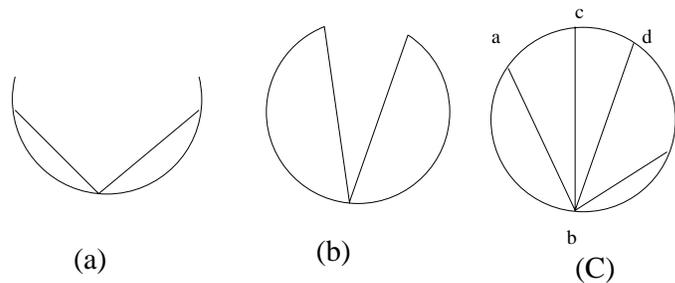


Figura 5: Estimado de curvaturas al fijar tres puntos en un círculo: (a) el ángulo entre vectores es grande, por lo tanto la aproximación en este caso es bastante buena. (b) ángulo muy pequeño, no es muy bueno como aproximación de curvatura a través de los tres puntos. (c) Aún cuando la distancia entre punto no es la misma y están dispuestos en diferentes posiciones con aparente curvatura distinta, no lo es así en estos casos, donde varias ternas de puntos sobre el círculo tienen la misma curvatura

### 3.4 Algoritmo Greedy

En esta sección se presentará un algoritmo Greedy el cual permite a un contorno, con términos de continuidad de primer y segundo orden, converger sobre un área de alta energía de imagen, en este caso, aristas. Este algoritmo permite la inclusión de restricciones duras como las descritas por Amini[AAT88], pero es mucho más rápido que su algoritmo de orden  $O(nm^3)$ , siendo en este caso de orden  $O(nm)$  para un contorno que tenga  $n$  puntos, a los cuales se les permite moverse a cualquier punto en una vecindad de tamaño  $m$  en cada iteración. Aunque el algoritmo no garantiza el dar un mínimo global, los resultados son comparados con otros métodos.

La cantidad a ser minimizada por este algoritmo es:

$$E = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{image}) ds \quad (9)$$

La forma de esta ecuación es similar a la ecuación ( 3). El primer y segundo término son las restricciones de continuidad de primer y segundo

orden. Estas se corresponden a la energía interna de la ecuación ( 3). El último término mide ciertas cantidades de imágenes tales como extensión de una arista o intensidad y, es la misma que en la ecuación ( 3). Ningún término para restricciones externas fue incluido, aunque sería posible hacerlo. Los parámetros  $\alpha$ ,  $\beta$  y  $\gamma$  son usados para balancear la influencia relativa de los tres términos.

El algoritmo propuesto es iterativo, como lo son los de Kass y Amini. Durante cada iteración, una vecindad de cada punto es examinada, y el punto en la vecindad dando el valor más pequeño para el término de energía es elegido como una nueva localización del punto. Solamente contornos cerrados son considerados aquí, por lo tanto todos los índices aritméticos son módulos  $n$ .

En los ejemplos que se muestran más adelante:  $\alpha=1$ ,  $\beta$  igual a 0 o 1 (dependiendo si se asume una esquina en esa localización), y  $\gamma=1.2$ . Esto fue elegido para que el gradiente de imagen tuviese una ligera mayor importancia que los otros términos de continuidad, para determinar cuando un punto del contorno se mueve.

Determinar una apropiada aproximación para el primer término en la ecuación ( 3), presenta algunas dificultades. Usando  $|\mathbf{v}_i - \mathbf{v}_{i-1}|^2$  hará que la curva se encoja, puesto que esta condición realmente minimiza la distancia entre puntos. Esto también contribuye al problema de agrupamiento de puntos sobre porciones fuertes del contorno. Estos efectos también se dan en el algoritmo Greedy, donde cada punto es movido de acuerdo a consideraciones locales. El comportamiento para estos puntos es que siempre serán movidos cerca de los puntos previos, los cuales también se mueve al punto más distante desde el punto siguiente. Esto causa una reacción en cadena, moviendo todos los puntos hacia el primero. Observando el comportamiento del algoritmo dado, es evidente que un término que facilite la distribución uniforme de los puntos reflejaría el comportamiento deseado del contorno mucho más que uno que hace que este se encoja. Así, el algoritmo presentado por Williams utiliza la diferencia entre la distancia promedio,  $\bar{d}$ , y la distancia entre los dos puntos en consideración:  $\bar{d} - |\mathbf{v}_i - \mathbf{v}_{i-1}|$ . De esta forma, los puntos que tienen la distancia próxima al promedio tendrán el valor mínimo. El valor es normalizado al dividirlo por el valor más grande en la vecindad en la que el punto puede moverse, dando un valor en  $[0,1]$ . Al final de cada iteración, se calcula un nuevo valor de  $\bar{d}$ .

El segundo término en la ecuación ( 9) es la curvatura. Ya que la formulación del término de continuidad hace que los puntos estén uniformemente espaciados,  $|\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}|^2$ , la fórmula en la columna 4 de la tabla 1, da un estimado razonable de la curvatura multiplicada por una constante. El término constante no es significativo puesto que este término, como el término de continuidad, es normalizado al dividirlo por el mayor valor en la vecindad, dando un número en  $[0,1]$ . Esta fórmula tiene la ventaja que es computacionalmente más eficiente que las discutidas previamente.

El tercer término en la ecuación ( 9),  $E_{image}$ , es la fuerza de imagen, la cual es la magnitud del gradiente. La magnitud del gradiente en cada punto de imagen es un entero de 8 bits con valores entre 0-255. Existe una diferencia significativa entre un punto con magnitud de gradiente de 240, y otro que tiene magnitud 255. Esto no está reflejado cuando los valores son normalizados al dividirlos por 255. Así, dando la magnitud en un punto (mag) y el gradiente máximo (max) y mínimo (min) en cada vecindad, la cantidad (max-mag)/(max-min) es usada para normalizar el término de la amplitud de aristas.

Este término de magnitud del gradiente es negativo, por lo tanto, puntos con magnitud gradiente grande tendrán valores pequeños. Otra condición a ser usada es que, si (max-min)<5 entonces min=max-5. Esto evita grandes diferencias en los valores de este término que provienen de áreas donde la magnitud gradiente es casi uniforme. Por ejemplo, cuando todos los puntos de una vecindad han sido examinados y tienen los valores 47, 48, y 49, el término de magnitud del gradiente sería 0, -0.5, -1.0 para puntos con, esencialmente, la misma magnitud gradiente. Así, un punto tendría una fuerte tendencia a permanecer fijo en un punto con magnitud gradiente 49, aunque esta no es un punto de arista fuerte, teniendo un mínimo de 5 en el denominador, daría -0.6, -0.8, -1.0 para el término gradiente, el cual refleja de manera más exacta la similitud entre puntos. Cerca de una arista está situación normalmente no surge, pero si el contorno tiene puntos que comienzan no muy lejos de la arista final o regiones expandidas donde existen gaps en la arista, los puntos del contorno se pueden resistir a moverse sin esta restricción.

Al final de cada iteración, se incluye un paso que determina la curvatura en cada punto del nuevo contorno, si el valor es un máximo de curvatura, se impone  $\beta_i = 0$  para la siguiente iteración. Este paso funciona como un proceso de bajo nivel dando un feedback al paso de minimización de la energía. La curvatura es calculada en cada uno de los  $n$  puntos por  $[\Delta x_i/\Delta s_i - \Delta x_{i+1}/\Delta s_{i+1}]^2 + [\Delta y_i/\Delta s_i - \Delta y_{i+1}/\Delta s_{i+1}]^2$ . Esta es la medida dada en la columna 5 de la tabla [1], la cual está relacionada al ángulo entre los vectores. Esta fórmula requiere mayor cálculo que la usada en el cálculo principal del algoritmo, pero se utiliza pocas veces, ( $n$ ), y se usa porque facilita el calculo de umbral. Los puntos que tienen una curvatura por encima de un treshold son considerados puntos esquinas para la siguiente iteración. Una condición adicional para asignar que un punto es una esquina es que la magnitud del gradiente debe estar por encima de algún valor mínimo. Esto evita que se formen esquinas hasta que el contorno esté próximo a una arista. En resumen,  $\beta$  es 0 en los puntos que satisfacen las condiciones anteriores: curvatura máxima por encima de un umbral de curvatura y por encima de un umbral de gradiente. Esto permite formar una esquina y reduce la curvatura entre estos puntos.

La función de energía es calculada para la ubicación de  $v_i$  en cada uno de

sus vecinos. La que tiene el menor valor es elegida como una nueva ubicación de  $v_i$ . Como  $v_{i-1}$  ya ha sido movido a su nueva posición durante la presente iteración, esta ubicación es usada con la de cada una de las ubicaciones propuestas para  $v_i$  para calcular el término de continuidad de primer orden. La ubicación de  $v_{i+1}$  aún no ha sido movida. Esta ubicación, junto con  $v_{i-1}$ , es usada para calcular la restricción de segundo orden para cada punto en la vecindad de  $v_i$ . Para  $i = 0$ , solo se usan valores iniciales. Por esta razón  $v_0$  es procesado dos veces, una como el primer punto en la lista y una vez más como el último punto. En la figura(6.b) se muestra como trabaja el algoritmo.

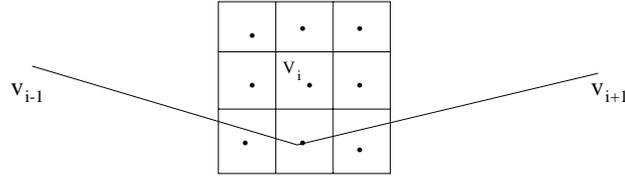


Figura 6: La función de energía es calculada en cada  $v_i$  y en cada uno de sus 8 vecinos. Los puntos anteriores y posteriores a dicho punto del contorno, son usados en el cálculo de la restricción de continuidad. La posición que tiene el mínimo valor de energía es elegida como una nueva posición de  $v_i$

Seguidamente se da el pseudo código del Algoritmo Greedy.

**Índice aritmético es módulo  $n$ .**

**Inicializar**  $\alpha_i, \beta_i$  y  $\gamma_i, \forall i$ .

*/\* ciclo para mover los puntos a una nueva posición \*/.*

**para**  $i = 0$  **hasta**  $n$ .

$E_{min} = \text{BIG}$

**para**  $j = 0$  **hasta**  $m - 1$

$E_j = \alpha_i E_{cont,j} + \beta_i E_{curv,j} + \gamma_i E_{image,j}$

**Si**  $E_j < E_{min}$  **entonces**

$E_{min} = E_j$

$j_{min} = j$

*/\* mover el punto  $v_i$  a la posición  $j_{min}$ \*/*

*/\* contador para los puntos que se mueven\*/*

**Si**  $j_{min}$  **no es la posición actual, entonces,**  $ptsmoved + = 1$

*/\* el proceso determina donde existirá una esquina en la siguiente iteración\*/*

**Para**  $i = 0$  **hasta**  $n - 1$

$c_i = \left| \frac{\bar{\mathbf{u}}_i}{|\bar{\mathbf{u}}_i|} - \frac{\bar{\mathbf{u}}_{i+1}}{|\bar{\mathbf{u}}_{i+1}|} \right|^2$

**Para**  $i = 0$  **hasta**  $n - 1$

*/\* Si la curvatura es mayor que sus vecinos\*/*

**Si**  $c_i > c_{i-1}$  **y**  $c_i > c_{i+1}$

*/\* y la curvatura es mayor que un umbral\*/*

**Y si**  $c_i > threshold1$

/\* y la fuerza de arista esta por encima de un umbral\*/  
**Y**  $\text{mag}(\mathbf{v}_i) > \text{umbral2}$   
/\* relaja la curvatura en el punto  $i^*$ \*/  
**Entonces**  $\beta_i = 0$   
**Hasta que**  $\text{:pstmoved} < \text{umbral3}$

### 3.5 Resultados de los Contornos Activos

Williams mostró el procedimiento del algoritmo descrito en la sección previa con los resultados para el algoritmo Greedy junto con la solución original con cálculo Variacional, y el algoritmo de Programación Dinámica. Estos programas fueron ejecutados sobre una imagen sintética, un Cuadrado, (fig.7), y tres imágenes reales: Cubo (fig.8), botella (fig.9),y taza (fig.10). La imagen de la taza probó el comportamiento de los algoritmos cuando el contorno rodea una región donde las aristas fuesen débiles o inexistentes. El contorno inicial en el Cubo fue producido por el algoritmo de enlace de aristas desarrollado por Williams and Shah [DM90]. En todas las imágenes, los puntos sobre el contorno que satisfacen la condición de curvatura grande son marcados con un cuadrado grande. En estos puntos, la restricción de segundo orden de continuidad fue relajada. La vecindad examinada en cada punto consistía en el punto mismo y sus 8 vecinos. De esta forma, el tamaño de la vecindad  $m$ , fue 9. La imagen (7.a) muestra el comienzo del contorno, con 40-60 puntos espaciados a una distancia aproximada de 4-6 pixeles. El umbral para  $\beta = 0$  fue 0.25. Resultados experimentales confirmaron que este valor era suficientemente grande para determinar cuando se obtiene una esquina o una línea curva. El umbral para la magnitud de gradiente mínimo antes de una esquina fue marcado como 100, cuando la magnitud del gradiente estaba en el rango  $[0,255]$ . Esto no es un valor crítico, y una amplitud en el rango de valores da resultados similares. Un último umbral fue el número de puntos movidos para determinar que la convergencia se ha producido. Pequeños valores no nulos entre 2 y 5 para este umbral resultaron buenos.

La parte (b) muestra los resultados de permitir al contorno original converger alrededor de las aristas del objeto usando el método de Cálculo Variacional propuesto por Kass.

La parte (c) muestra el resultado del algoritmo de Programación Dinámica para las cuatro imágenes. Para reducir la tendencia al agrupamiento en puntos fuertes del contorno, una de las restricciones prohíbe movimiento perpendicular en la dirección de gradiente máximo. Esto no evita que los puntos no se introduzcan o infiltren sobre la arista, moviéndose desde puntos de aristas fuertes, los cuales no fueron los puntos más próximos a la ubicación actual, sino que provienen del movimiento de aristas a lo largo del contorno para puntos más grandes una vez que estos han alcanzado dicha arista. El movimiento a lo largo del contorno también extiende el tiempo

de convergencia cuando esta restricción no es incluida. El umbral dado fue el número de puntos que se movieron durante la iteración. Usualmente el número de puntos que son movidos en cada iteración disminuye cuando el contorno se aproxima a la ubicación de la arista.

La parte (d) en cada figura muestra el resultado del algoritmo Greedy. Los resultados obtenidos con los tres métodos anteriores son similares, uno da ligeramente mejores resultados en una imagen, mientras un método diferente da mejores resultados en otra imagen. Las esquinas no son colocadas con el método de Kass, por lo tanto este da un contorno que está más redondeado en las esquinas. Los resultados para la caja son casi idénticos para los tres algoritmos, siendo mejor en la arista superior izquierda con el algoritmo Greedy, mientras la arista superior derecha es ligeramente mejor con Programación Dinámica. En la imagen de la botella, dos puntos están muy cerca en la tapa y en el lado derecho con el algoritmo de Programación Dinámica, pero resulta uniformemente espaciado en el algoritmo Greedy a causa de la forma distinta de la restricción de primer orden de continuidad. Los puntos de aristas no siguen el cuello de la botella tan bien como en el algoritmo Greedy. Como se esperaba, el contorno no prosigue bien del lado derecho de la taza con ninguno de los métodos. Cuando las aristas de la taza no son fuertes, los puntos pertenecientes al background parecen ser los puntos de convergencia con el algoritmo Greedy. Los otros tres métodos convergen a las aristas de sombras del lado derecho de la esquina de la base de la taza, puesto que fue la primera arista encontrada cuando el contorno se aproximaba a la misma.

La tabla [2] da el número de puntos en cada contorno, el umbral usado para la convergencia, el tiempo empleado en segundos, el número de iteraciones requeridos para converger, y el número de puntos de curvatura grande en el cual la restricción de continuidad de segundo orden fue relajada, colocando  $\beta$  en 0. Ningún método fue significativamente mejor en el número de iteraciones requeridos. En el cuadrado y la taza se tienen pocas iteraciones con el algoritmo Greedy, mientras la caja y la botella requirieron pocas iteraciones con el algoritmo de Programación Dinámica. Los resultados de contornos obtenidos con el algoritmo Greedy son al menos tan buenos como los de Programación Dinámica, y el tiempo de ejecución es mucho mejor. El enfoque de Calculo Variacional requiere un tiempo similar al método Greedy para cada iteración, pero usualmente converge en pocas iteraciones. Si fuese permitido cambiar los valores de  $\beta$  entre iteraciones, sería necesario calcular la inversa de una matriz pentadiagonal  $n \times n$  disminuyendo su velocidad.

Las figuras (11) y (12) muestran el uso de las snake para aplicaciones de segmentación de imágenes.

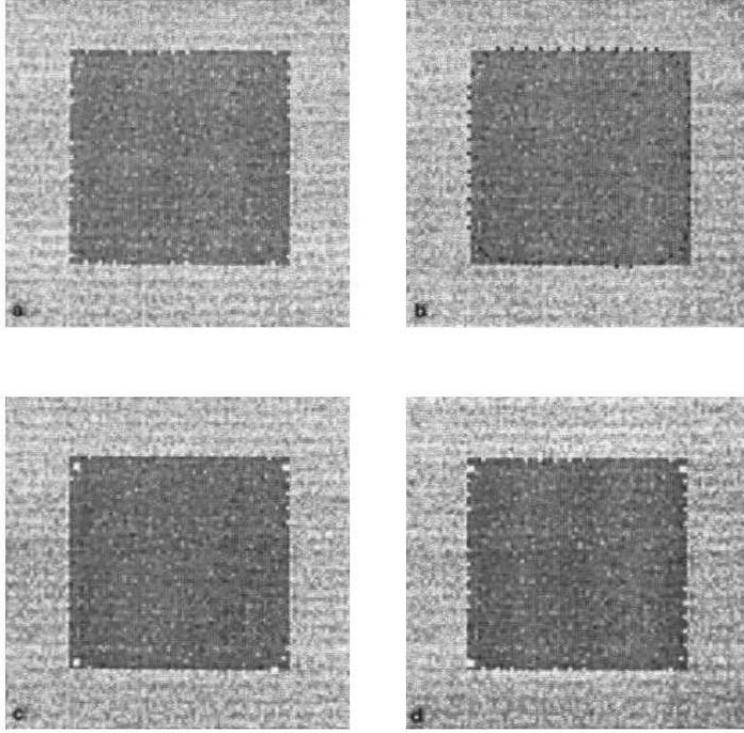


Figura 7: Cuadrado:(a)Contorno original, (b)Método de Kass, (c)Algoritmo de programación Dinámica, (d)Algoritmo Greedy.

## 4 Mallas Activas 2D

Una Active Net puede ser descrita como una placa activa definida por  $\mathbf{v}(r, s) = (x(r, s), y(r, s))$ , donde  $(r, s) \in ([0, 1] \times [0, 1])$ . Esta parametrización describe una placa bidimensional actuando en un mundo bidimensional, en este caso una imagen.

Para controlar una Active Net, se define una función de energía:

$$E(\mathbf{v}) = \int_0^1 \int_0^1 E_{int}(\mathbf{v}(r, s)) + E_{ext}(\mathbf{v}(r, s)) dr ds \quad (10)$$

donde  $E_{int}$  es la energía interna de la Net la cual controla la forma y la estructura de la Net y  $E_{ext}$  es la energía debida a fuerzas externas.

### 4.1 Energía interna

El término de la energía interna está definido por:

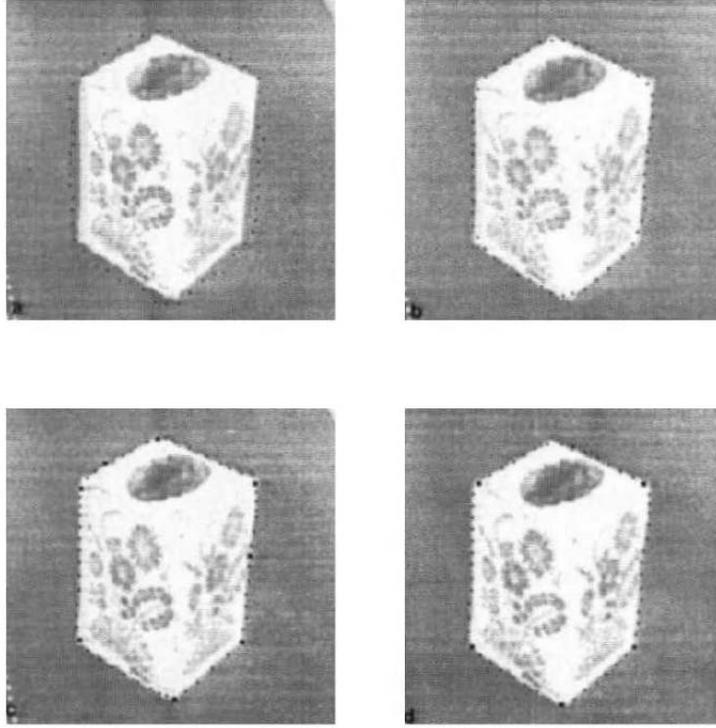


Figura 8: Caja:(a)Contorno original, (b)Método de Kass, (c) Algoritmo de programación Dinámica, (d)Algoritmo Greedy.

$$E_{int}(\mathbf{v}) = (\alpha(|\mathbf{v}_r(r, s)|^2 + |\mathbf{v}_s(r, s)|^2) + (\beta(|\mathbf{v}_{rr}(r, s)|^2 + 2|\mathbf{v}_{rs}(r, s)|^2 + |\mathbf{v}_{ss}(r, s)|^2))) \quad (11)$$

donde los subíndices indican derivadas parciales y  $\alpha$  y  $\beta$  son los coeficientes de primer y segundo orden para controlar la "suavidad" de la malla.

Esta energía interna,  $E_{int}$ , define un estabilizador para controlar la continuidad [D.T86] que asegura una malla continua  $C^1$ . La suma de las primeras derivadas parciales son proporcionales a la desviación de energía en una membrana y las segundas derivadas parciales reflejan la energía necesaria para que una placa se torne curva o se pueda doblar. En la práctica, las primeras derivadas hacen que la malla se pueda contraer y las segundas derivadas asegura la suavidad y rigidez de la misma.

La definición de energía interna en la ecuación ( 11) es continua en el modelo paramétrico. Para calcular la energía computacionalmente, el dominio paramétrico  $[0,1] \times [0,1]$  es discretizado como una cuadrícula definida por nodos internos espaciados  $(k, l)$ . Por lo tanto se emplea una formulación en diferencias finitas del término de la energía interna:

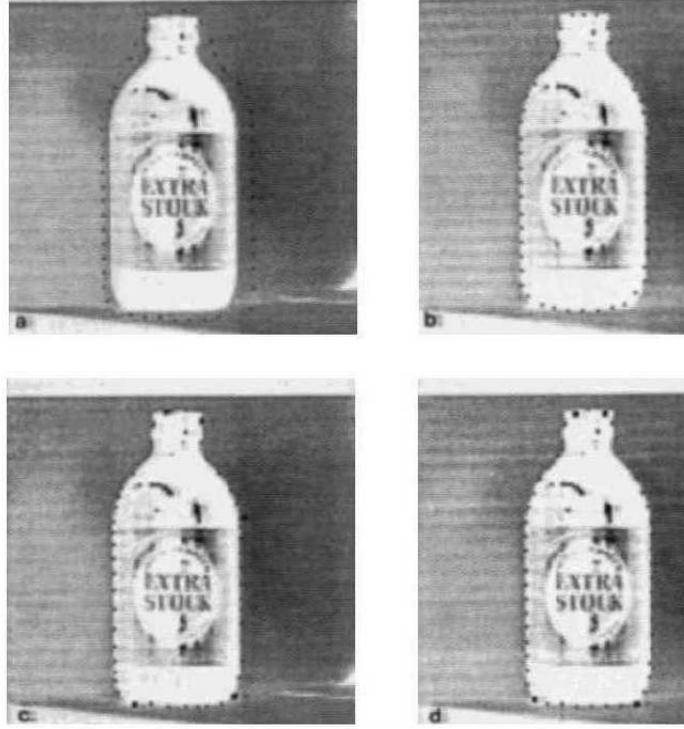


Figura 9: Botella:(a)Contorno original, (b)Método de Kass, (c)Algoritmo de programación Dinámica, (d)Algoritmo Greedy.

$$\begin{aligned}
 \mathbf{d}_r^+(r, s) &= (\mathbf{v}(r + k, s) - \mathbf{v}(r, s))/k \\
 \mathbf{d}_r^-(r, s) &= (\mathbf{v}(r, s) - \mathbf{v}(r - k, s))/k \\
 \mathbf{d}_s^+(r, s) &= (\mathbf{v}(r, s + l) - \mathbf{v}(r, s))/l \\
 \mathbf{d}_s^-(r, s) &= (\mathbf{v}(r, s) - \mathbf{v}(r, s - l))/l
 \end{aligned} \tag{12}$$

Donde  $\mathbf{d}^+$  y  $\mathbf{d}^-$  son las diferencias hacia adelante y hacia atrás respectivamente.

Puesto que se utiliza un esquema de actualización local para la minimización de la energía, los estimados centrales de las derivadas de primer orden no se usan. Con un estimado central, la contribución del nodo actual no es tomado en cuenta, dejando solamente las contribuciones de los nodos vecinos. En el algoritmo Greedy, la minimización de la energía es realizada al mover el nodo actual a los sitios vecinos y calcular la energía del nodo en esos sitios. Pero el nodo actual no es usado en el cálculo del estimado central y, por lo tanto, la derivada de primer orden no cambia.

Para disponer de información tanto de los estimados hacia adelante y hacia atrás para ser usados en el cálculo de la primera derivada y evitar

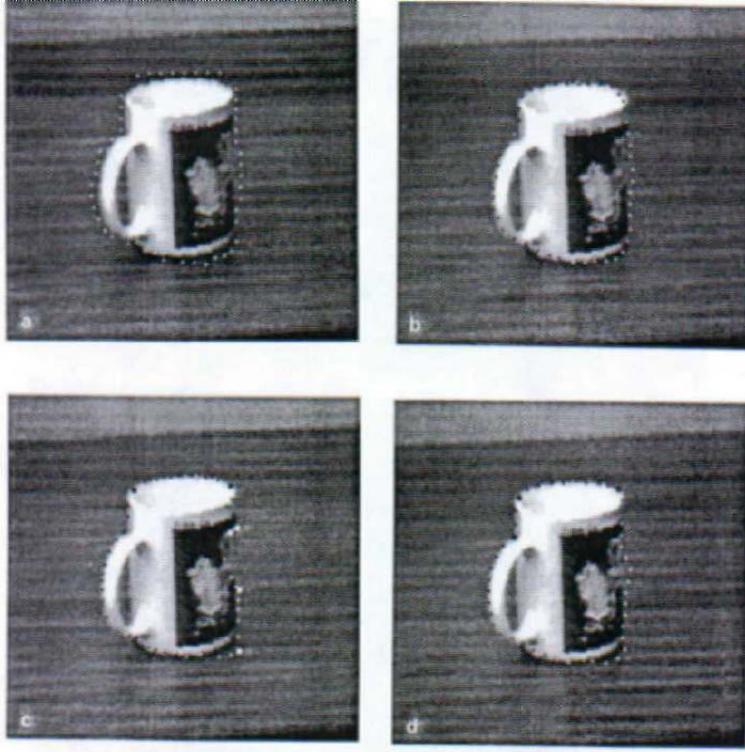


Figura 10: Taza: (a) Contorno original, (b) Método de Kass, (c) Algoritmo de programación Dinámica, (d) Algoritmo Greedy.

estimados centrales, se usarán las siguientes formulaciones:

$$\begin{aligned} |(\mathbf{v}_r(r, s))^2| &= (|\mathbf{d}_r^+(r, s)|^2 + |\mathbf{d}_r^-(r, s)|^2)/2 \\ |(\mathbf{v}_s(r, s))^2| &= (|\mathbf{d}_s^+(r, s)|^2 + |\mathbf{d}_s^-(r, s)|^2)/2 \end{aligned} \quad (13)$$

Las segundas derivadas son estimadas usando la técnica de diferencias finitas estándar:

$$\begin{aligned} \mathbf{v}_{rr}(r, s) &= (\mathbf{v}(r - k, s) - 2\mathbf{v}(r, s) + \mathbf{v}(r + k, s))/k^2 \\ \mathbf{v}_{ss}(r, s) &= (\mathbf{v}(r, s - l) - 2\mathbf{v}(r, s) + \mathbf{v}(r, s + l))/l^2 \\ \mathbf{v}_{rs}(r, s) &= (\mathbf{v}(r - k, s) - \mathbf{v}(r - l, s + l) - \mathbf{v}(r, s) + \mathbf{v}(r, s + l))/kl \end{aligned} \quad (14)$$

## 4.2 Energía Externa

$E_{ext}$  es la energía de las fuerzas externas que actúan sobre la malla. Estas fuerzas son usualmente fuerzas de imágenes diseñadas para atraer la malla hacia las características del objeto. También se pueden aplicar fuerzas de restricciones externas tales como fuerzas de gravedad o fuerzas aplicadas por el usuario. En este contexto, solo se usarán fuerzas de imagen.

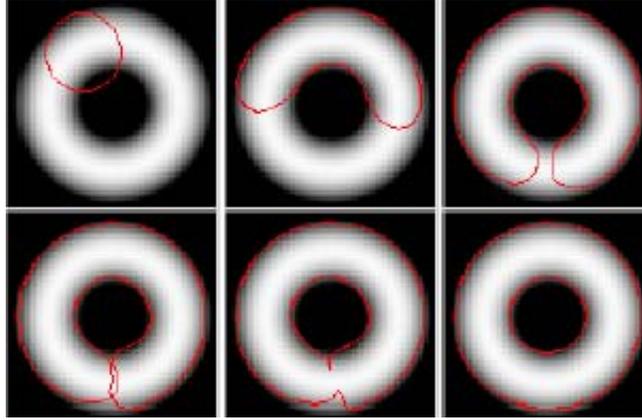


Figura 11: Segmentación de un objeto en forma de anillo por un contorno activo.

			AG		P.D			C.V	
imagen	tamaño	seg	iter	esqui	seg	iter	esqui	seg.	iter
cuadrado	58	0.25	2	3	12.162	3	4	0.35	4
caja	56	1.867	15	5	25.157	6	8	0.466	4
botella	50	1.217	11	4	29.465	8	5	1.499	12
taza	49	0.7	7	3	34.153	10	6	0.3	4

Tabla 2: Comparación del tiempo de ejecución, número de iteraciones, y número de discontinuidades de segundo orden realizados con los métodos Alg. Greedy (AG), Prog. Dinámica (PD) y Cálculo Variacional (CV)

Por lo tanto la energía externa puede ser escrita como:

$$E_{ext}(\mathbf{v}(r, s)) = f[I(\mathbf{v}(r, s))] \quad (15)$$

donde  $f$  es una función general de la intensidad de la imagen  $I(v)$ . Tsumiyama y otros [YKK89], en su trabajo original, usó la siguiente función para la energía externa

$$E_{ext}(\mathbf{v}(r, s)) = f[I(\mathbf{v}(r, s)), V_{9 \times 9}(\mathbf{v}(r, s))] \quad (16)$$

donde  $V_{9 \times 9}(v(r, s))$  es la variación local calculada en una ventana  $9 \times 9$ . Lamentablemente, esta medida de energía puede ser atrapada por pequeñas zonas en la imagen, puesto que la medida de la energía solo depende de la intensidad de la imagen en un solo nodo. Para facilitar la propagación de la información de la imagen en una malla activa, Yoshino y Kawashima [KTY93] sugirieron una nueva medida de energía:

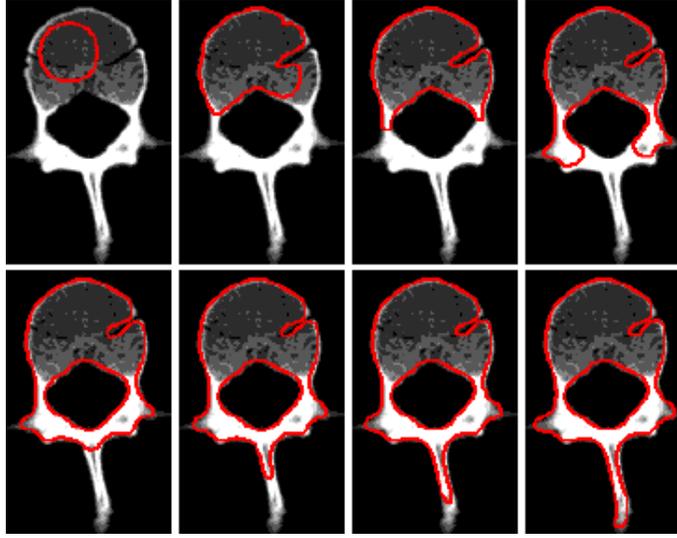


Figura 12: Convergencia de un contorno activo hacia una imagen, obtenida por CT-scan, de una vértebra.

$$E_{ext}(\mathbf{v}(r, s)) = \omega f[I(\mathbf{v}(r, s))] + \frac{\rho}{\mathcal{N}(r, s)} \sum_{\mathbf{p} \in \mathcal{N}(r, s)} \frac{1}{\|\mathbf{v}(r, s) - \mathbf{v}(\mathbf{p})\|} f[I(\mathbf{v}(\mathbf{p}))] \quad (17)$$

$$f[I(\mathbf{v}(r, s))] = \overline{g[I(\mathbf{v}(r, s)_3)]}$$

donde  $\overline{[I(\mathbf{v}(r, s)_3)]}$  es la intensidad media en una ventana 3x3,  $g$  es una función general y  $\mathcal{N}(r, s)$  es el conjunto de nodos  $(r, s)$  "4-vecinos".  $\omega$  y  $\rho$  son coeficientes de ponderación. Esta función de energía promedia la energía de imagen de sus nodos vecinos por la inversa de la distancia. Esto hace que la malla activa propague fuerzas de imagen a través de la malla y asegure una deformación más estable.

Filtrando primero la imagen usando  $\overline{[I(\mathbf{v}(r, s)_3)]}$  hace que los nodos sean movidos fuera de regiones "puntiagudas" y pequeñas de la imagen. Estos objetos son "suavizados" y las fuerzas internas en una malla activa pueden moverse a través de ella. Lamentablemente esto también reduce la precisión de localización de aristas de nodos fronteras. Para los nodos interiores, el tamaño del filtro de suavización es determinado basándose en el tamaño de la imagen. Usualmente un filtro 5x5 es usado para una imagen de medidas 200x200 pixeles.

La función  $g$  determina características de la imagen, las cuales generan fuerzas de atracción y repulsión.  $g$  está definida de forma diferente para nodos interiores y nodos fronteras. Asumiendo que el objeto pueda ser

más brillante que el resto de la imagen, los nodos internos serían atraídos hacia áreas de la imagen más brillantes. El papel fundamental de los nodos fronteras es parar la contracción de la malla activa en la frontera de un objeto. Por lo tanto,  $g$  está definida inversamente por los nodos fronteras, es decir, los nodos fronteras son atraídos hacia áreas oscuras y repulsados de áreas brillantes.

En conclusión, la medida de la energía externa sería:

$$\begin{aligned}
 E_{ext}(\mathbf{v}(r, s)) &= \omega f[I(\mathbf{v}(r, s))] \\
 &\quad + \frac{\rho}{\mathcal{N}(r, s)} \sum_{\mathbf{p} \in \mathcal{N}(r, s)} \frac{1}{\|\mathbf{v}(r, s) - \mathbf{v}(\mathbf{p})\|} f[I(\mathbf{v}(\mathbf{p}))] \\
 f[I(\mathbf{v}(r, s))] &= \begin{cases} h[I_{max} - \overline{I(\mathbf{v}(r, s)_n)}] & \text{para nodos internos} \\ h[I(\mathbf{v})] & \text{para nodos fronteras} \end{cases}
 \end{aligned} \tag{18}$$

donde  $I_{max}$  es el máximo de intensidad de imagen y  $h$  es una función de escalamiento apropiada .

En algunos casos, especialmente con imágenes de bajo contraste, el hacer un filtrado a la imagen, mejora lo resultados. Un problema particular ocurre cuando los nodos fronteras se "deslizan" o atraviesan la frontera del objeto. Bro-Nielsen comenta que usando una versión de la imagen, donde previamente se ha aplicado un umbral, como "input" para los nodos fronteras reduce este problema considerablemente. Otra solución es poner los pesos de la energía externa mayor para los nodos frontera.

### 4.3 Minimización de la Energía

La minimización de la energía se realiza localmente usando una versión mejorada del Algoritmo Greedy [DM92] . Cada nodo en el modelo activo es probado sucesivamente en las posiciones de los pixeles en una "8-vecindad" del nodo. Para cada posición, se determina la contribución de energía del nodo y se mueve el nodo a la posición que minimiza la energía local:

$$E_{local}(\mathbf{v}(r, s)) = E_{int}(\mathbf{v}(r, s)) + E_{ext}(\mathbf{v}(r, s)) \tag{19}$$

Aunque el Algoritmo Greedy originalmente es rápido y razonablemente robusto, tiene debilidades relacionadas al esquema de actualización local. Es sensible al mínimo local y las actualizaciones sucesivas de los nodos pueden causar una rotación no deseada o movimientos del modelo. En lo que sigue se discute este problema, haciendo un análisis en paralelo entre el algoritmo de Campos Aleatorios de Markov y el algoritmo del Modo Iterado Condicional (ICM) de Besag[J.B86]. El algoritmo ICM está justificado dentro de las bases o estructuras estadísticas.

### 4.3.1 Formulación Estocástica

En primer lugar queremos resaltar que en este apartado se harán mención y se usarán resultados de probabilidades, principalmente a los teoremas asociados a Campos Aleatorios de Markov, teorema de Hammersley, teorema de Bayes y algunas definiciones de sistemas de vecindades como cliques, entre otros. Todas estas herramientas se utilizan para obtener una mejora del algoritmo de minimización de la energía usando el algoritmo Greedy. Puede ser recomendable una lectura del apéndice (al final del trabajo) para una mejor comprensión de lo que sigue.

Consideremos el conjunto de nodos de una malla activa como un Campo Aleatorio de Markov (MRF)

[J.B86] relacionado al sistema  $\mathcal{N}_4$  de “4-vecinos”. Usando el teorema Hammersley-Clifford [D.G88] podemos escribir la distribución a priori, como una distribución de Gibbs:

$$p(\mathbf{v}) = \frac{1}{Z_1} \exp(-U_1(\mathbf{v})) \quad (20)$$

donde  $Z_1$  es una constante de normalización.

$U_1$  es una función de energía que suma los potenciales  $V_c(v)$  de todos los cliques (ver apéndice)  $c \in \mathcal{C}$  en el sistema de vecindad.

$$U_1(\mathbf{v}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{v}) \quad (21)$$

con los cliques definidos en la figura(13), la función de energía puede ser definida usando la función de energía definida en la ecuación ( 11):

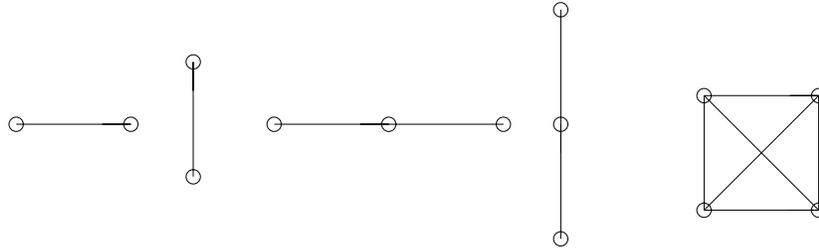


Figura 13: *Cliques en una rejilla regular*

$$U_1(\mathbf{v}) = \sum_{(r,s)} E_{int}(\mathbf{v}(r,s)) \quad (22)$$

si el modelo de observación  $p(I|\mathbf{v})$  está definido como la distribución de Gibbs:

$$\begin{aligned} p(I|\mathbf{v}) &= \frac{1}{Z_2} \exp(-U_2(\mathbf{v})) \\ U_2(\mathbf{v}) &= \sum_{(r,s)} E_{ext}(\mathbf{v}(r,s)) \end{aligned} \quad (23)$$

luego, la distribución posterior se puede escribir usando el teorema de Bayes:

$$\begin{aligned} p(\mathbf{v}|I) &= \frac{p(I|\mathbf{v})p(\mathbf{v})}{p(I)} \\ &= \frac{1}{Z} \exp(-U_1(\mathbf{v}) - U_2(\mathbf{v})) \end{aligned} \quad (24)$$

donde  $Z$  es la combinación de  $Z_1$ ,  $Z_2$  y  $p(I)$ .

Teniendo definida la distribución posterior, un estimado Máximo A Posteriori (MAP) puede ser determinado usando el algoritmo Modo Iterado Condicional (ICM) [J.B86].

Cada nodo de la malla activa  $(r,s)$  es visitado por turno. Para cada nodo, una nueva posición  $\hat{\mathbf{v}}(r,s)$  es elegida para minimizar la expresión

$$\begin{aligned} &p(\mathbf{v}(r,s)|\mathbf{I}(r,s), \hat{\mathbf{v}}(r,s) \quad \forall \mathbf{p} \neq (r,s)) \\ &\propto p(\mathbf{I}(r,s)|\mathbf{v}(r,s)) \times p(\mathbf{v}(r,s)|\hat{\mathbf{v}}(\mathbf{p}), \forall \mathbf{p} \in \mathcal{N}(r,s)) \\ &\propto \exp(-U_1(\mathbf{v}(r,s)) - U_2(\mathbf{v}(r,s))) \end{aligned} \quad (25)$$

para  $\mathbf{v}(r,s)$ , donde  $\hat{\mathbf{v}}(r,s)$  es el estado de los demás nodos. Minimizar (25)

es lo mismo que minimizar:

$$U_1(\mathbf{v}(r,s)) + U_2(\mathbf{v}(r,s)) = \sum_{(r,s)} E_{int}(\mathbf{v}(r,s)) + E_{ext}(\mathbf{v}(r,s)) \quad (26)$$

la cual es la energía local  $E_{local}$  de la ecuación (19) que minimiza el algoritmo Greedy. En conclusión, existe una clara similitud entre el algoritmo Greedy y el algoritmo ICM. Producto de esta similitud, las consideraciones relacionadas al algoritmo ICM también deben, intuitivamente, aplicarse al algoritmo Greedy.

### 4.3.2 Mejoras deducidas del ICM

El trabajo de Basag [J.B86] se notó que la convergencia del algoritmo ICM solo se asegura si los nodos locales son actualizados en una forma asíncrona. Pero se puede tener una convergencia "razonable" si los nodos son visitados en un orden tal que ellos no se solapen unos a otros con sus vecinos. Por lo

tanto se define un camino  $(r, s) = \mathbf{w}(i)$  tal que este camino visita todos los nodos sobre una senda sin solapamiento.

Para una malla activa, las pruebas mostraron que el mejor camino es aquel que visita los nodos en forma espiral. Primero los nodos en la frontera exterior son visitados en un círculo, luego los nodos en la frontera de los nodos interiores restantes, etc. En cada anillo, los nodos son visitados saltando de uno a otro. Puesto que los nodos son visitados de esta manera, estos nodos no se solapan unos a otros con sus vecinos. Desafortunadamente, los nodos en anillos vecinos pueden solaparse. En la practica el camino en espiral acelera la convergencia, ya que moviéndose primero por los anillos exteriores, libera la tensión para los anillos internos y permite que estos se muevan más rápidos. De este modelo de camino surge lo siguiente:

MEJORA 1. visitar los nodos en forma espiral de afuera hacia adentro, desde la frontera, saltando de un nodo a otro en cada anillo.

Aunque mejorado por la inclusión del camino, el algoritmo Greedy aun tiende a fallar en el mínimo local, probablemente a causa de la actualización local. Para resolver este problema, una segunda mejora inspirada por el método de optimización Simulated Annealing [SD84], ha sido usado.

MEJORA 2. Mover los nodos que no han sido movidos de ninguna manera, con una probabilidad  $p$  a una posición seleccionada aleatoriamente en una "8-vecindad".

Un nodo, que no es movido por el algoritmo durante una visita local, será movido con probabilidad  $p$  hacia un vecino arbitrario seleccionado de manera aleatorio.  $p = 5$  es una buena elección.

#### 4.4 Separación de una malla activa 2D

El permitir que una malla activa se separe o divida fue sugerido inicialmente por Yoshino y otros [13][KTY93]. En ese trabajo, ellos sugieren el modelado de arcos en la malla como resortes, y desenlazar la conexión cuando la tensión en estos resortes excedan un valor umbral predefinido. La longitud de un arco fue usada como una medida de la tensión, ya que la tensión en un resorte es proporcional a su longitud. Pero Bro-Nielsen[M.B94] comenta que dicho criterio no es suficiente. Un claro ejemplo se muestra en la figura (14). En la imagen izquierda el arco más extendido está en la arista curvada del objeto de la derecha. En vez de seleccionar un arco en la porción obvia, que es la mitad de la malla, la regla sugerida por Yoshino corta este arco más extendido.

Otro problema no considerado por Yoshino es la topología en la secuencia de cortes. El método sugerido por Yoshino selecciona arcos para ser cortados independientemente. En el ejemplo de la figura (14), la mejor sucesión de cortes sería una que siguiera una línea recta a través de la mitad de la malla entre los dos objetos. Una dependencia entre arcos subsiguientes sería por lo tanto necesaria. A continuación, se presenta un nuevo algoritmo para cortar

arcos en una malla activa 2D que resuelve estos problemas.

En vez de usar la longitud de un arco como criterio para cortar las uniones entre vértices, se utiliza la distancia del arco desde el nodo más próximo colocado en el objeto de la imagen. El algoritmo contiene los siguientes pasos.

**Algoritmo de corte 1**

Si la cantidad de nodos movidos en la última iteración fue menor que  $p_{change}$ , entonces hacer:

1. Etiquetar todos los nodos como OBJETO o NO-OBJETO basado sobre un umbral de la intensidad de la imagen en cada nodo.
2. Para todos los nodos, determinar la distancia  $dist(\mathbf{p})$  a los nodos OBJETO más próximos, usando una distancia transformada sobre la malla activa.
3. Cortar el arco de la malla, entre los nodos vecinos  $\mathbf{p}_1$  y  $\mathbf{p}_2$  para los cuales la distancia media  $(dist(\mathbf{p}_1)+dist(\mathbf{p}_2))/2$  es mayor y más grande que una distancia mínima  $dist_{min}$ .

El cálculo de la distancia transformada puede ser realizado usando cualquier método disponible. Los requerimientos de distancia mínima aseguran que los arcos en la frontera de un objeto no se cortan. Este algoritmo resuelve el problema de seleccionar un arco apropiado para ser cortado, pero los cortes subsiguientes aún no son interdependientes.

Para hacer una sucesión de cortes dependientes, el registro de la sucesión debe ser salvado y usado en la malla en forma flexible y eficiente. La solución es etiquetar los arcos próximos a un arco que se corta con una cierta prioridad. Esta prioridad controla la selección de arcos para una sucesión de cortes. Dando una alta prioridad para arcos en la dirección del corte, la dirección de corte puede ser controlada. El algoritmo final, incluyendo estas prioridades es:

**Algoritmo de corte 2.** Inicialización antes de la minimización de la energía: Colocar la prioridad de nodos interior y frontera en 0 y 1 respectivamente.

Después de cada iteración: Si la proporción de nodos cambiados en la última iteración es menor que  $p_{change}$ , entonces hacer:

1. Etiquetar todos los nodos como OBJETO o NO-OBJETO, basado en un umbral de la intensidad de la imagen en el nodo.
2. Para cada nodo, determinar la distancia  $dist(\mathbf{p})$  a los nodos OBJETO más próximos usando una distancia transformada sobre la malla activa.
3. Cortar el arco de la malla entre los nodos vecinos  $\mathbf{p}_1$  y  $\mathbf{p}_2$  para el cual:

- (a) La prioridad es mayor
  - (b) Si dos arcos tienen igual prioridad, seleccionar el arco con la mayor distancia promedio  $(dist(\mathbf{p}_1) + dist(\mathbf{p}_2))/2$ .
  - (c) La distancia de los nodos  $dist(\mathbf{p}_1)$  y  $dist(\mathbf{p}_2)$  es mayor que una distancia mínima  $dist_{min}$ .
4. Si un arco se corta: Colocar la prioridad de todos los nodos en  $MIN(1, pr)$  donde  $pr$  es la prioridad de un nodo. Marcar el siguiente arco en la dirección del corte con 3. Marcar los arcos frontera conectado al arco de prioridad 3 con prioridad 2.

Este algoritmo asegura que el corte comienza en un apropiado arco central en una malla activa y sigue un camino directo razonable. La colocación de prioridades se muestran en la figura(15).

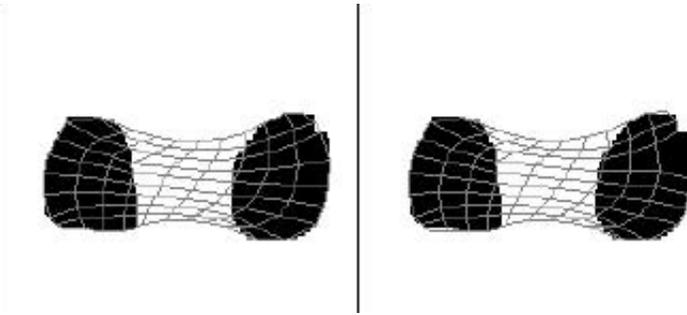


Figura 14: Corte de malla no apropiado

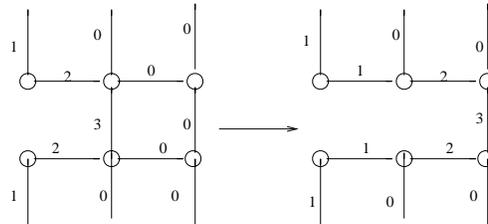


Figura 15: Colocación de prioridades después que un arco es cortado

## 4.5 Resultados Experimentales

Estos comentarios son clásicos y están basados en el trabajo de Bro-Nielsen.

El algoritmo de separación puede ser usado para extraer algunos objetos de una misma escena. Ver figura(16). Estos muestran el comportamiento

general del algoritmo de división o separación. Nótese que en ambos casos la separación ocurre en cortes razonables y las sub-mallas resultantes son bien estructuradas y suaves. Algunas irregularidades pueden ser notadas a lo largo de la frontera de las sub-mallas. estas irregularidades están en las esquinas de la malla, estas no pueden ser suavizadas completamente. Morten Bro-Nielsen [M.B94] comenta que en el trabajo de Sakaue [MK91] se usa un malla circular. Este tipo de malla resuelve algunos de los problemas concernientes a las esquinas, pero definir división o separación en tales mallas es sumamente complicado.

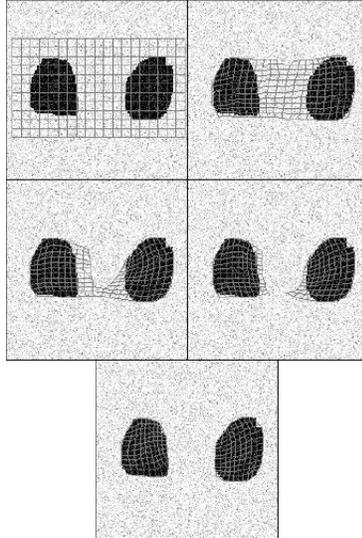


Figura 16: Prueba del algoritmo de división de una malla. En las imágenes del medio se ve como se va efectuando la división de la malla activa, y en la última imagen se aprecia el resultado final de esta división de la malla en dos sub-mallas.

## 5 Cubos Activos

La extensión de mallas activas 2D a cubos activos 3D es fácil y solo involucra el agregar una dimensión. Por esta razón, esta sección presenta la teoría en forma corta.

Un Active Cube, o cubos activos, es un cubo sólido volumétrico, modelando el comportamiento de un “dado” elástico. La parametrización es  $\mathbf{v}(r, s, t) = (x(r, s), y(r, s), t(r, s))$ , donde  $(r, s, t) \in ([0, 1] \times [0, 1] \times [0, 1])$  y el cubo activo es controlado por una función de energía definida por :

$$E(\mathbf{v}) = \int_0^1 \int_0^1 \int_0^1 E_{int}(\mathbf{v}(r, s, t)) + E_{ext}(\mathbf{v}(r, s, t)) dr ds dt \quad (27)$$

Donde  $E_{int}$ , energía interna del cubo, controla la forma y estructura de la malla y  $E_{ext}$  es la energía debido a fuerzas externas.

La energía está definida como:

$$\begin{aligned}
E_{int}(\mathbf{v}) = & (\alpha(|\mathbf{v}_r(r, s, t)|^2 + |\mathbf{v}_s(r, s, t)|^2 + |\mathbf{v}_t(r, s, t)|^2)) \\
& + (\beta(|\mathbf{v}_{rr}(r, s, t)|^2 + |\mathbf{v}_{ss}(r, s, t)|^2 + |\mathbf{v}_{tt}(r, s, t)|^2)) \\
& + (\gamma(2|\mathbf{v}_{rs}(r, s, t)|^2 + 2|\mathbf{v}_{rt}(r, s, t)|^2 + 2|\mathbf{v}_{st}(r, s, t)|^2))
\end{aligned} \tag{28}$$

los subíndices indican derivadas parciales y  $\alpha$ ,  $\beta$  y  $\gamma$  son coeficientes de primero y segundo orden que controlan la "suavidad" de la malla. El comportamiento del Active Cube debido al término de primero y segundo orden es similar al caso bidimensional, los cuales indican la contracción y rigidez respectivamente. La fuerza de energía externa está definida por:

$$\begin{aligned}
E_{ext}(\mathbf{v}(r, s)) = & \omega f[I(\mathbf{v}(r, s, t))] \\
& + \frac{\rho}{\mathcal{N}(r, s, t)} \sum_{\mathbf{p} \in \mathcal{N}(r, s, t)} \frac{1}{\|\mathbf{v}(r, s, t) - \mathbf{v}(\mathbf{p})\|} f[I(\mathbf{v}(\mathbf{p}))] \\
f[I(\mathbf{v}(r, s, t))] = & \begin{cases} h[I_{max} - \overline{[I(\mathbf{v}(r, s, t)_n)}]] & \text{para nodos internos} \\ h[I(\mathbf{v})] & \text{para nodos fronteras} \end{cases}
\end{aligned} \tag{29}$$

donde  $\overline{[I(\mathbf{v}(r, s, t)_n)]}$  es la intensidad media en un cubo  $n \times n$ ,  $I_{max}$  es la intensidad máxima de la imagen y  $h$  es una apropiada función de escalado. La minimización de la energía es realizada usando la versión mejorada del Algoritmo Greedy. Excepto por el mayor tiempo de cómputo, a causa del mayor número de nodos, el Active Cube es muy similar a las mallas activas.

El mayor tiempo de cálculo puede ser reducido fijando la componente  $z$  de  $\mathbf{v} = (x, y, z)$ . En este caso, solo se permite que los nodos se muevan dentro de un corte fijado del objeto, como dentro de un plano, pero no entre cortes. Esto usualmente reduce el tiempo de cálculo a la mitad.

Otra aceleración puede ser realizada cuando a la imagen se le aplica un umbral previamente. En este caso, una distancia transformada puede ser aplicada a las áreas NO-OBTEJO, midiendo la distancia al objeto desde cada pixel. Esto generará un campo de atracción para nodos en el cubo, permitiendo que estos sigan la dirección del campo gradiente, y por lo tanto convergerá más rápido.

## 5.1 Resultados Experimentales

Como una aplicación de estos Cubos Activos, Bro-Nielsen los empleó para obtener la cabeza de un paciente con los datos obtenida por CT-scan. La figura(17) muestra el Active Cube después de contraerse alrededor del scan

CT de una cabeza. La normal de la superficie no está determinada correctamente con el software usado para visualizar la superficie. La ligera deformación del lado derecho de la boca es causada por un tubo que fue insertado en la boca del paciente durante la adquisición de la imagen.

La figura(18)muestra el resultado del interior de la malla activa 3D. La estructura regular de los nodos en el interior se percibe claramente. Nótese que los nodos no se mueven entre cortes, puesto que la componente  $z$  fue fijada durante este experimento.

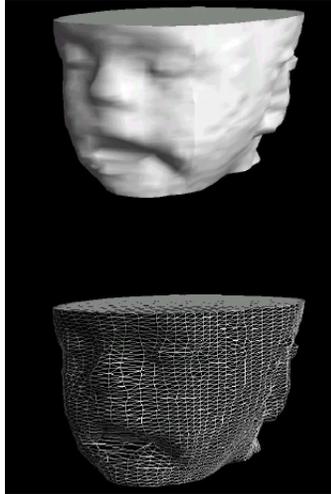


Figura 17: Resultado al aplicar el algoritmo de cubo activo a un CT-scan de una cabeza.

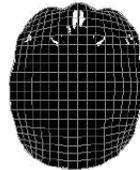


Figura 18: Interior de un Active net 3D.

## 5.2 Cubos Activos en simulación quirúrgica

Un tema de gran interés en años recientes es la aplicación de tecnología informática que permita al cirujano realizar operaciones médicas simuladas sobre modelos 3D de un paciente. Esta metodología tiene la ventaja de la práctica quirúrgica antes de hacer operaciones difíciles y analizar los resultados de la operación. Usando esta técnica es posible examinar los efectos del movimiento de la mandíbula sobre una vista general del rostro del paciente.

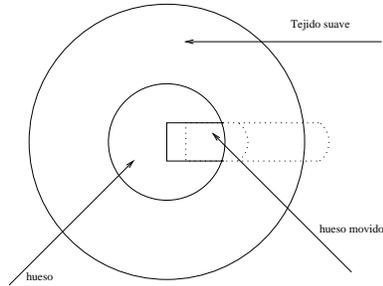


Figura 19: Objeto a ser simulado usando Active Cube.

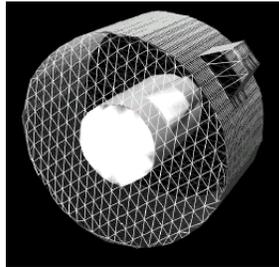


Figura 20: Modelado del objeto y el movimiento de tejido duro usando cubos activos.

Hasta ahora las operaciones simuladas han sido realizadas usando modelos de superficies 3D del paciente. La desventaja de este enfoque es que el interior del paciente no está disponible en el modelo. En consecuencia, no es posible cortar la superficie y ver el interior del modelo. El movimiento de tejido o hueso bajo la superficie tampoco es posible.

Bro-Nielsen propone el uso del modelo sólido activo 3D como el modelo básico para operaciones simuladas. La ventaja de este modelo es que los nodos interiores definen el interior del modelo. Estos nodos permiten al cirujano cortar sobre el modelo o mover huesos con un efecto realista, ya



Figura 21: Resultado final, para el movimiento de tejido suave, usando cubo activo.

que el interior del modelo está incluido en el modelo. Los párrafos siguientes describen un experimento usando Active Cube para modelar movimientos de huesos.

La figura(19) muestra una ilustración del objeto de prueba usado en este experimento. Este objeto simulará una parte de la pierna. La operación simulada consiste en mover hacia afuera una parte de hueso de la pierna. El movimiento intuitivo del tejido suave no es un movimiento rígido sino una deformación elástica que resulta en un bulto sobre la superficie del tejido suave, es decir la piel. Para modelar este comportamiento, se usa un Active Cube que simula las propiedades elásticas del tejido suave. El enfoque es:

1. Deformar el Active Cube para emparejar o ajustar el objeto antes de la operación simulada.
2. Inicializar el Active Cube con la forma resultante. Mantener el Active Cube sin ninguna energía externa se transforma en obtener la forma inicial.
3. Etiquetar todos los nodos en el Active Cube como tejidos HARD o SOFT, dependiendo sobre el estado del voxel que ocupa.
4. Realizar el movimiento modelado del hueso moviendo los nodos del Active Cube que solapan el volumen movido.
5. Mantener los nodos del Active Cube etiquetado como tejido SOFT, permitiendo que estos encuentren la posición de equilibrio sujeto a restricciones de los nodos de tejido HARD.
6. Después de la convergencia, el Active Cube resultante es un modelo de deformación física del paciente

El almacenamiento del modelo del Active Cube en el item ( 1) es realizado usando un tensor métrico  $\mathcal{G}$  en cada nodo. Este tensor métrico es una matriz 3x3 definida por:

$$G_{ij}(\mathbf{v}(\mathbf{p})) = \frac{\partial \mathbf{v}}{\partial \mathbf{x}_i} \frac{\partial \mathbf{v}}{\partial \mathbf{x}_j} \quad (30)$$

donde  $\mathbf{x}_i$  es la  $i$ -ésima componente del vector  $(x, y, z)$ . El tensor métrico almacena el modelo de un sólido definido por distancias y ángulos.

La figura(20) muestra una visualización de la malla activa 3D con el movimiento inicial del tejido suave debido a la operación simulada del hueso. En la figura(21) se ha dejado que la malla activa converja a una forma de equilibrio del modelo. La figura muestra como la superficie se ha movido elásticamente en respuesta a una presión hacia afuera del hueso. Como resultado, se ve un bulto suave sobre la superficie del tejido en contraste al movimiento rígido del tejido que se muestra en la figura(20).

## 6 Conclusiones

Este trabajo hace la revisión de varios modelos activos junto con sus respectivos enfoques de solución. En General, la cantidad de nodos involucrados en estos modelos activos es grande. Usualmente, cuando un contorno activo tiene  $4n$  nodos, las mallas activas tienen  $n^2$  nodos, donde los nodos extras están distribuidos en el interior del contorno activo. El número de nodos es, en ambos casos, grande y aumenta rápidamente con el tamaño del modelo activo.

Como una aplicación de estos modelos activos, un ejemplo claro se mostró en la sección previa. Un Active Cube deformado se usó como un modelo de estructura básico para una operación simulada. Esta operación simulada se distingue por modelar el comportamiento real 3D del tejido suave cuando parte de tejido rígido debajo de este, es movido. Este tejido es modelado usando los nodos y las conexiones del cubo activo conjuntamente con un modelo de propiedades elásticas del tejido suave. Como resultado, es posible percibir el cambio de la superficie del pseudo - paciente como un cambio en la posición del hueso bajo la superficie.

Se mostró la utilidad de dividir mallas activas, la cual puede ser usada como una buena herramienta de segmentación de escenas con varios objetos. Se presentó el algoritmo fundamental para estos modelos activos, snake, con algunos comentarios sobre los métodos de empleados por varios investigadores en esta área. Además de mejoras interesantes de dicho algoritmo, en beneficio de mejores tiempo de ejecución, robustez, interacción y exactitud, no lograda en su totalidad con ningún método, ya que siempre se tienen prioridades y solo algunas de estas características deseables son alcanzadas. También se mostraron ventajas y desventajas de las técnicas usadas para la minimización de la funcional de la energía.

Considero que unas de las líneas a seguir trabajando es en el estudio de las energías internas y externas. Con variaciones sobre el spline de las fuerzas externas, y atacar la solución de la funcional de energía desde el punto de vista de análisis numérico, aunque esto es costoso en tiempo de ejecución. Pero se ganaría también en robustez. También se podría seguir explotando el framework probabilístico, con lo cual se ganaría velocidad de cálculo.

Entre otras cosas a seguir en esta línea de investigación sería tratar, por ejemplo, con objetos de una escena con “perforaciones o huecos”, lo cual representaría un área de imagen conexa pero no simplemente conexa, y tratar de obtener una representación del objeto en términos de una malla activa, que pudiese ajustarse a dicho objeto, pero en la cual no estén representados nodos de la malla en zonas o áreas que no pertenezcan al objeto. Esto se lograría rompiendo la malla en las zonas que no pertenecen al objeto por medio de fuerzas externas definidas sobre el funcional de energía, y que luego la malla se plegaría a la frontera interior de estas zonas. En el caso de

un toro, se obtendría la representación del mismo por medio de una malla activa, sin que existan nodos de la malla dentro del agujero del toro. De esta manera, tendríamos no solo la representación del toro en términos de su frontera, que podría lograrse con un contorno activo cuando el mismo es colocado dentro del objeto y por medio de fuerzas externas (fuerzas de inflado) es ajustado al borde del toro, si no una representación volumétrica

Obtener esta representación sería útil pues podría ser usada en aplicaciones médicas, por ejemplo. Con la representación de estas zonas con huecos en términos de puntos internos se podría emplear en la representación del corazón (ventrículo izquierdo). Pues una vez obtenida esta representación volumétrica, se podrían aplicar fuerzas interactivas sobre puntos de control bien definidos (los de la malla 3D como una extensión de la malla activa 2D) para simular bombeo de sangre.

## 7 Apéndice

### 7.1 Conceptos elementales de probabilidades y estadística

En este apartado se presentan brevemente algunas definiciones y proposiciones de utilidad en este trabajo.

Hablaremos un poco de probabilidad y estadística.

La idea intuitiva de probabilidad es de un número entre 0 y 1 que está asociado con la verosimilitud de que un suceso ocurra. Es 0 cuando estamos seguros que el suceso no va a ocurrir y 1 cuando estamos seguros que sí ocurrirá. La probabilidad a veces se estima por la frecuencia relativa, por ejemplo, una manera de aproximarnos a la probabilidad de que una intervención quirúrgica arriesgada tenga éxito es consultar el registro de intervenciones quirúrgicas realizadas sobre pacientes similares, si de las últimas 10, 8 han sido un éxito, la frecuencia relativa es  $8/10=0.8$ , que será su probabilidad o se parecerá a ella. La frecuencia relativa cambia, en el ejemplo anterior, si el registro, en lugar de 10 pacientes tuviera 11, la frecuencia relativa sería necesariamente distinta ( $8/11$  ó  $9/11$ ), pero hay una ley empírica que establece que cuando el número de ensayos (por ejemplos pacientes) es suficientemente grande, la frecuencia relativa se estabiliza. A veces se define la probabilidad como el límite de la frecuencia relativa. (Problema: ¿Cómo saber en cada caso, si el número de ensayos es suficientemente grande?).

Hay situaciones en las que se puede calcular la probabilidad: sí todos los resultados del experimento son igualmente probables, entonces la probabilidad se define (definición de Laplace) como el cociente entre el número de casos favorables y el número de casos totales.

$$p(S) = \frac{CF}{CT}$$

Formalización:

**Definición 7.1 (Experimento aleatorio).** *Experimento que puede ser repetido bajo “las mismas condiciones”, del que puede establecerse el conjunto de sus posibles resultados, pero no puede predecir un resultado concreto.*

**Definición 7.2 (Espacio muestral).** *Conjunto de posibles resultados.*

**Definición 7.3 (Suceso).** *Cualquier subconjunto del espacio muestral.*

Si se representa el espacio muestral por  $\mathbf{W}$  y los sucesos por  $\mathbf{A}$ :  $\mathbf{A} \subset \mathbf{W}$ . Dado que el conjunto vacío es subconjunto de cualquier conjunto,  $\emptyset \subset \mathbf{W}$ , y todo conjunto es subconjunto de sí mismo  $\mathbf{W} \subset \mathbf{W}$ . Así, tanto el espacio muestral como el conjunto vacío son sucesos.

**Definición 7.4 (Probabilidad).** *Sea  $\mathbf{W}$ : espacio muestral,  $\mathbf{P}(\mathbf{W})$  partes, o conjunto de sucesos, o álgebra de sucesos. se define la probabilidad, o función de probabilidad, a cualquier función  $p$  que cumpla los siguientes axiomas:*

1.  $p(\mathbf{A}) \geq 0, \forall \mathbf{A} \subset \mathbf{W}$
2.  $p(\mathbf{A}_1 \cup \mathbf{A}_2 \cup \mathbf{A}_3 \cup \dots) = p(\mathbf{A}_1) + p(\mathbf{A}_2) + p(\mathbf{A}_3) \dots$
3. si  $p(\mathbf{A}_1 \cap \mathbf{A}_2 = \emptyset \quad \forall i \neq j$
4.  $p(\mathbf{W}) = 1$ .

A la estructura  $(\mathbf{W}, \mathbf{P}(\mathbf{W}), p)$  se le llama *espacio de probabilidad*.  
Propiedades de la probabilidad.

1.  $p(\mathbf{A}^c) = 1 - p(\mathbf{A})$ , donde  $\mathbf{A}^c$  es el suceso complementario de  $\mathbf{A}$ .
2.  $\mathbf{A}_1 \subset \mathbf{A}_2 \Rightarrow p(\mathbf{A}_1) \leq p(\mathbf{A}_2)$
3.  $p(\mathbf{A}) \leq 1$
4.  $p(\mathbf{A} \cup \mathbf{B}) = p(\mathbf{A}) + p(\mathbf{B}) - p(\mathbf{A} \cap \mathbf{B})$ .(Regla de la adición.)

Probabilidad condicionada.

Como la probabilidad está ligada a nuestra ignorancia sobre los resultados de la experiencia, el hecho de que ocurra un suceso, puede cambiar la probabilidad de los demás.

La probabilidad de que ocurra un suceso  $\mathbf{A}$  si ha ocurrido un suceso  $\mathbf{B}$  se denomina y se define por:

**Definición 7.5 (Probabilidad condicionada).**

$$p(\mathbf{A}/\mathbf{B}) = \frac{p(\mathbf{A} \cap \mathbf{B})}{p(\mathbf{B})} \quad \text{si } p(\mathbf{B}) \neq 0.$$

**Definición 7.6.** Dos sucesos son independientes si y solo si  $p(\mathbf{A} \cap \mathbf{B}) = p(\mathbf{A})p(\mathbf{B})$ . En ese caso

$$p(\mathbf{A}|\mathbf{B}) = \frac{p(\mathbf{A} \cap \mathbf{B})}{\mathbf{B}} = \frac{p(\mathbf{A} \cdot \mathbf{B})}{\mathbf{B}} = p(\mathbf{A})$$

Reglas de la probabilidad.

**Definición 7.7.** Se llama una partición a un conjunto de sucesos  $\mathbf{A}_i$  tales que  $\mathbf{A}_1 \cup \mathbf{A}_2 \cup \dots \cup \mathbf{A}_n = \mathbf{W}$  y  $\mathbf{A}_i \cap \mathbf{A}_j = \emptyset \quad \forall i \neq j$ .

**Definición 7.8 (Regla de probabilidad total).** Si un conjunto de sucesos  $\mathbf{A}_i$  forman una partición del espacio muestral y  $p(\mathbf{A}_i) \neq 0 \quad \forall \mathbf{A}_i$ , entonces

$$p(\mathbf{B}) = \sum_{i=1}^n p(\mathbf{B}|\mathbf{A}_i)p(\mathbf{A}_i).$$

**Teorema 7.1 (de Bayes).** Si los sucesos  $\mathbf{A}_i$  son una partición y  $\mathbf{B}$  un suceso tal que  $p(\mathbf{B}) \neq 0$ , entonces

$$p(\mathbf{A}_i|\mathbf{B}) = \frac{p(\mathbf{B}|\mathbf{A}_i)p(\mathbf{A}_i)}{\sum_{j=1}^n p(\mathbf{B}|\mathbf{A}_j)p(\mathbf{A}_j)} \quad \text{para } i = 1 \dots n.$$

A continuación algunas definiciones de estadística.

**Definición 7.9 (Estadística).**

- Conjunto de métodos científicos ligados a la organización, recopilación, presentación y análisis de datos, tanto para la deducción de conclusiones como para la toma de decisiones razonables de acuerdo con tales análisis.
- Arte de la decisión en presencia de incertidumbre.

En ciertas áreas de científicas aparece el concepto de experimento aleatorio (experimento que repetido en las mismas condiciones no produce el mismo resultado) y asociado al mismo el de Variable Aleatoria.

Una variable aleatoria está caracterizada por la llamada función densidad de probabilidad, a partir de la cual se obtiene la probabilidad para sus posibles valores para cada condición. Los objetivos de la investigación científica se pueden entender en términos de encontrar y describir las variables de interés y las relaciones entre ellas para el problema en estudio.

Así, la estadística es la ciencia que estudia los métodos que permiten realizar este proceso para variables aleatorias. Y la descripción completa de una variable aleatoria está dada por su función densidad de probabilidad (*fdp*). Cada *fdp* está caracterizada usualmente por dos parámetros: *media* y *varianza*.

**Definición 7.10 (Variable aleatoria (v.a)).** *Función que asocia un número real, perfectamente definido, a cada punto muestral.*

Una v.a puede ser continua o discreta. Las v.a permiten definir la probabilidad como una función numérica (de variable real) en lugar de una función de conjunto.

Por ejemplo: Tiramos una moneda 3 veces. Representamos cara por  $c$  y cruz por  $z$ .

$$\mathbf{W} = ccc, ccz, czc, zcc, czz, zcz, zzc, zzz.$$

La probabilidad de cada suceso elemental es  $1/8$ , puesto que la probabilidad de sacar cara en una tirada es  $1/2$  según la definición clásica y las tiradas son independientes. Definimos la variable aleatoria  $\mathbf{X}$ : número de caras, que pueden tomar los valores  $0,1,2,3$ . Se buscan todos los puntos muestrales que dan lugar a cada valor de la variable y a ese valor se le asigna la probabilidad del suceso correspondiente. A esta función se le denomina *función densidad de probabilidad*. En el caso de variables discretas, como en el ejemplo cara - cruz, es una función que para cada valor de la variable da su valor de probabilidad, así, se tiene que  $f(\mathbf{X})$  tomaría los valores  $1/8, 3/8, 3/8, 1/8$ . Para variables continuas la probabilidad de que una variable tome cualquier valor concreto es  $0$ , por lo tanto la *fdp* solo permite calcular la probabilidad para un intervalo del tipo  $a < \mathbf{X} < b$ , mediante el área bajo la curva de la *fdp*.

**Definición 7.11 (Esperanza o media).** *Se define la Esperanza como:*

$$\mu_X = E[X] = \begin{cases} \sum_X Xf(X) & \text{caso discreto} \\ \int_{-\infty}^{\infty} Xf(X) dX & \text{caso continuo.} \end{cases}$$

Si  $\mathbf{X}$  es una v.a cualquier función de ella,  $h(\mathbf{X})$ , es también una variable aleatoria, en consecuencia también se define este parámetro para una función de v.a.

$$\mu_h = E[h(X)] = \begin{cases} \sum_X h(X)f(h(X)) & \text{caso discreto} \\ \int_{-\infty}^{\infty} h(X)f(X) dX & \text{caso continuo.} \end{cases}$$

**Definición 7.12 (Varianza).** *Se define Varianza como:*

$$\sigma_X^2 = E[(X - \mu_X)^2]$$

Aunque para el cálculo se usa la siguiente fórmula equivalente.

$$\sigma_X^2 = E[X]^2 - \mu_X^2.$$

La varianza mide la dispersión de la variable alrededor de la media.

## 7.2 Preliminares a los Campos Aleatorios de Markov y Distribución de Gibbs

Los campos Aleatorios de Markov son una parte de la Teoría de la Probabilidad que proporciona una herramienta para analizar dependencias espaciales o contextuales de fenómenos físicos. También se usan, entre otras aplicaciones, en el problema del etiquetado en visión para establecer las distribuciones de probabilidad de las etiquetas. Muchos problemas en visión pueden ser considerados como problemas de etiquetado, de manera que la solución a dicho problema es el conjunto de etiquetas asignados a los pixeles de la imagen. Por esta razón, este problema es una representación natural para estudiar los Campos Aleatorios de Markov.

Un problema de etiquetado puede especificarse en términos de un conjunto de estados y un conjunto de etiquetas. Sea  $S$  un conjunto discreto con  $m$  estados,  $S = \{1, 2, \dots, m\}$ .

Un estado puede representar un punto o una región del espacio cartesiano. El conjunto de estados puede ser clasificado en términos de su homogeneidad. Podemos denotar por

$$S = \{(i, j) / 1 \leq i, j \leq n\}.$$

al conjunto de localizaciones de los pixeles de una imagen 2D de tamaño  $n \times n$ .

Sea  $L$  el conjunto de etiquetas.  $L$  puede ser un conjunto discreto o continuo. En el caso continuo, el conjunto de etiqueta puede ser todos los reales o cualquier intervalo. En el caso discreto, las etiquetas se toman dentro de un conjunto discreto de  $M$  valores,  $L_d = \{l_1, \dots, l_m\} = \{1, \dots, M\}$ .

Una característica esencial de un conjunto de etiquetas es poder establecer un orden, así, en un conjunto discreto podemos ordenar los elementos en términos de la intensidad de los tonos de gris  $0, 1, \dots, 255$ . El problema de etiquetado consiste en asignar una etiqueta del conjunto  $L$  a cada estado de  $S$ . Por ejemplo, la detección de bordes en una imagen consiste en asignar una etiqueta  $f_i$  dentro del conjunto  $L = \{\text{borde}, \text{no borde}\}$  para cada  $i \in S$ , donde  $S$  está formado por los pixeles de la imagen.

**Definición 7.13 (configuración o etiquetado).** *Sea*

$$f = \{f_1, \dots, f_m\}.$$

donde a cada estado de  $S$  se le ha asignado una etiqueta de  $L$ . Y se define la siguiente función  $f$  como:

$$f : S \rightarrow L.$$

tal que a cada estado  $S$  le asigna una etiqueta  $f_i \in L$ .

Los estados  $S$  están relacionados por un sistema de vecinos.

**Definición 7.14.** Se llama sistema de vecinos para el conjunto de estado  $S$  a

$$N = \{N_i / \forall i \in S\}.$$

donde  $N_i$  es el conjunto de los estados vecinos a  $i$  para los que

$$i \notin N_i \quad i \in N_j \iff j \in N_i.$$

Para una rejilla regular  $S$ , el conjunto de vecinos de  $i$  está definido como el conjunto de estados próximos dentro de un radio  $r$

$$N_i = \{j \in S / [dist(pixel_j, pixel_i)]^2 \leq r, \quad j \neq i\}.$$

donde  $dist(A, B)$  es la distancia cartesiana entre  $A$  y  $B$ , y  $r$  es entero.

En el sistema de 4 vecinos de primer orden, llamado sistema de 4-vecinos, cada estado (interior) tiene cuatro vecinos, ver figura(22.a). En el sistema de vecinos de segundo orden, también llamado sistema 8-vecinos, hay ocho vecinos para cada estado figura(22.b). Los números  $n=1, \dots, 5$  mostrado en la figura(22.c) indican los estados de los vecinos de orden  $n$ .

El par  $\{S, N\}$  es un grafo en el sentido usual, donde  $S$  contiene los nodos y  $N$  determina las relaciones entre los nodos de acuerdo con el sistema de vecinos definido.

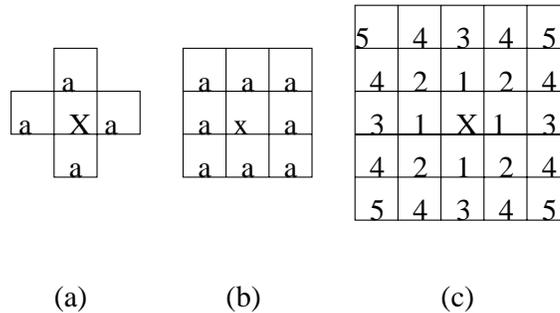


Figura 22: Vecinos en una rejilla regular  $S$ . (a) Sistema de primer orden o de 4-vecinos. (b) Sistema de segundo orden o de 8-vecinos. (c) Los números indican los estados vecinos exteriores de orden  $n = 1, \dots, 5$

**Definición 7.15 (Clique).** Una clique  $c$  para  $\{S, N\}$  se define como un subconjunto de  $S$  tal que  $c$  consiste en un único estado  $c = \{i\}$  o un par de estados vecinos  $c = \{i, j\}$  o tres estados vecinos  $c = \{i, j, k\}$  y así sucesivamente.

En otras palabras, un clique  $C$  es un estado (pixel o región) simple o un conjunto de estados tales que todos los estados que pertenecen a  $C$  son mutuamente vecinos.

Denotamos el conjunto de cliques de un solo estado, de dos estados, .... por  $C_1, C_2, \dots$  respectivamente, donde.

$$\begin{aligned}
 C_1 &= \{i/i \in S\} \\
 C_2 &= \{\{i, j\}/j \in N_i \quad i \in S\} \\
 C_3 &= \{\{i, j, k\}/i, j, k \in S \text{ son vecinos dos a dos}, \dots
 \end{aligned}$$

En la figura(23) se muestran diferentes tipos de cliques para vecinos de primer y segundo orden.

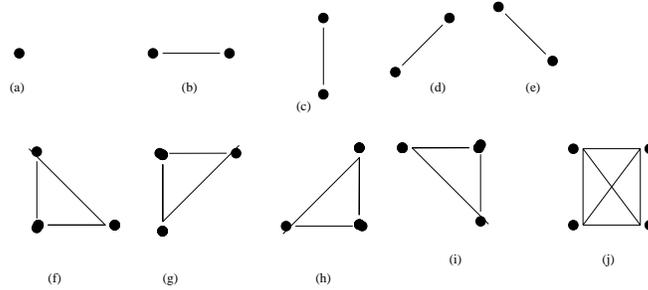


Figura 23: Cliques en una rejilla regular

### 7.3 Campos Aleatorios de Markov (CAM)

**Definición 7.16.** Sea  $F = \{F_1, \dots, F_m\}$  una familia de variables aleatorias definidas en  $S$ , donde cada variable aleatoria  $F_i$  toma un valor  $f_i \in L$ . Llamamos Campo aleatorio a la familia de variables aleatorias  $F$ . Se denota por  $F_i = f_i$  cuando el suceso  $F_i$  toma el valor  $f_i$ .

**Definición 7.17.** Un suceso conjunto  $\{F_1 = f_1, \dots, F_m = f_m\}$  se denomina una realización, y se denota por  $F = f$ . Se llama configuración de  $F$  a  $f = \{f_1, \dots, f_m\}$

Para un conjunto discreto de etiquetas  $L$ , se denota por  $p(F_i = f_i) = p(f_i)$  a la probabilidad de que la variable aleatoria  $F_i$  tome el valor  $f_i$  y se denota la probabilidad conjunta por  $p(F = f) = p(F_1 = f_1, \dots, F_m = f_m) = p(f)$

**Definición 7.18.** Una Familia de variables aleatorias  $F$  se dice que es un Campo Aleatorio de Markov sobre  $S$  respecto de  $N$  si y solo sí.

1.  $p(F = f) > 0 \quad \forall f \in L^m$
2.  $p(F_i = f_i | F_j = f_j, j \in S, j \neq i) = p(F_i = f_i | F_j = f_j, j \in N_i)$ .

La condición (1) exige que  $F$  sea un campo Aleatorio. La condición (2) se llama característica local. Se dice que la probabilidad de un suceso local a  $i$  condicionada a todos los sucesos restantes es equivalente a la probabilidad condicionada a los vecinos de  $i$ .

## 7.4 Distribución de Gibbs

**Definición 7.19.** *se dice que un conjunto de variables aleatorias  $F$  es un campo Aleatorio de Gibbs en  $S$  con respecto a  $N$  si y solo si sus configuraciones siguen una distribución de Gibbs. Una distribución de Gibbs tiene la siguiente función de densidad:*

$$p(f) = \frac{1}{Z} \exp\left(-\frac{1}{T} U(f)\right).$$

donde  $U(f)$  = función de energía.

$Z$  = constante de normalización.

$T$  = temperatura.

La función de energía viene dada por:

$$U(f) = \sum_c V_c(f).$$

$V_c$  son las cliques potenciales y la suma anterior está definida para todas las cliques posibles  $C$ . El valor de  $V_c(f)$  depende de la configuración local de la clique  $c$ . Se dice también que un Campo Aleatorio de Gibbs es homogéneo si  $V_c(f)$  es independiente de la posición relativa de la clique  $c$  en  $S$ .  $p(f)$  mide la probabilidad de la ocurrencia de una configuración determinada, o patrón  $f$ . Las configuraciones más probables son aquellas con energías más bajas.  $T$  controla la forma de la distribución. Cuando la temperatura es alta, todas las configuraciones tienden a estar igualmente distribuidas. Con  $T$  próximo a cero, la distribución se concentra alrededor de los mínimos globales de energía.

Un campo aleatorio de Markov se caracteriza por sus propiedades locales mientras que un campo aleatorio de Gibbs se caracteriza por sus propiedades globales (la distribución de probabilidad de Gibbs). El teorema de Hammersley-Clifford establece la equivalencia entre estos dos tipos de características.

**Teorema 7.2 (Hammersley-Clifford).**  *$F$  es un campo Aleatorio de Markov en  $S$  con respecto a  $N$  si y solo si  $F$  es un campo Aleatorio de Gibbs en  $S$  con respecto a  $N$ .*

la demostración del mismo se puede consultar en [J.B74], [J.M74], [RR80] entre otros.

## 7.5 Simulated Annealing

El Simulated Annealing es un método que explota la analogía entre, la manera en la cual un metal se enfría lentamente hasta alcanzar un mínimo en el estado de equilibrio de su estructura, y la búsqueda de un mínimo en un sistema más general.

La ventaja con respecto a otros métodos de optimización es su habilidad para evitar ser atrapado en un mínimo local. El algoritmo emplea una búsqueda aleatoria la cual no solo acepta cambios que hagan decreciente a la función objetivo,  $U(f) = U$  para simplificar, definida en el apartado distribución de Gibbs, sino también algunos cambios que la hacen creciente. Esto último es aceptado con una probabilidad

$$p = \exp\left(-\frac{\delta U}{T}\right)$$

donde  $\delta U$  es el incremento en  $U$  y  $T$  es un parámetro de control. Para este algoritmo se deben dar los siguientes elementos.

- Una representación de posibles soluciones.
- Un generador de cambios aleatorios en las soluciones.
- Una temperatura inicial y reglas para disminuir la temperatura en el proceso de búsqueda.

El algoritmo sería como sigue:

1. Asignar una temperatura inicial  $T$ . Asignar cada  $F_i$  de  $F$
2. Para cada  $F_i$  de  $f$ 
  - (a) Cambiar su etiqueta por otra cualquiera de  $L$
  - (b) Calcular  $\Delta U$
  - (c) Si  $\Delta U < 0$ , aceptar el valor actual de  $f$ . Sino, aceptar si  $\exp(-\Delta U/T) > \xi$ , donde  $\xi$  es un número generado aleatoriamente de una distribución uniforme en  $[0,1]$
3. Si la energía se estabiliza, salir. Sino, disminuir la temperatura en  $0 < k < 1$  de forma tal que  $T_{k+1} = kT$  y volver al paso 2.

## 7.6 Programación Dinámica

**Definición 7.20 (Programación Dinámica, PD).** *Método de resolver problemas por la combinación de soluciones de subproblemas.*

La programación dinámica, al igual que el método “divide y vencerás”, resuelve problemas por la combinación de soluciones de subproblemas ( programación en este contexto se refiere como un método tabulador). El algoritmo divide y vencerás parte el problema en subproblemas independientes, resuelve recursivamente los subproblemas, y luego combina sus soluciones para resolver el problema original. En contraste, la Programación Dinámica es aplicable cuando los subproblemas no son independientes, es

decir, cuando los subproblemas comparten sub-subproblemas. En este sentido, un algoritmo 'divide y vencerás' realiza más trabajo que el necesario, repetidamente resuelve los sub-subproblemas comunes. Mientras que la Programación Dinámica resuelve cada sub-subproblema exactamente una sola vez y luego guarda esta respuesta en una tabla, por esta razón evita el trabajo de re-calcular la respuesta cada vez que el subproblema es encontrado.

La programación es aplicada usualmente a problemas de optimización. Estos problemas pueden tener muchas posibles soluciones. En cada solución un valor, y deseamos encontrar una solución que optimice el valor mínimo (o máximo).

El desarrollo del algoritmo puede ser dividido en 4 pasos

1. Caracterizar la estructura de una solución óptima.
2. Definir recursivamente los valores de una solución óptima.
3. Calcular el valor de una solución óptima de la forma bottom-up (profundidad)
4. Construir una solución óptima a partir de la información.

Ventajas de PD

- Soluciona cada sub-subproblema exactamente una sola vez.
- Guarda soluciones parciales dentro de una tabla, lo que permite tener soluciones parciales.
- Tiene un menor costo de ejecución que los algoritmos recursivos.
- Puede tener ventaja de la superposición de subproblemas, variando un poco su funcionamiento.

Desventajas

- Necesita memoria para almacenar sus datos, por lo que para problemas grandes, es necesario una gran cantidad de memoria.
- Existe la posibilidad de solapamiento de subproblemas.

## Referencias

- [AAT88] A.Amini, A.Witkin, and T.Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *Second International Conference on Computer Vision*, pp.95-98, 1988.
- [DE80] D.Marr and E.Hildreth. Theory of edge detection. In *Proc. R. Soc. London B - 207*, 187-217, 1980.

- [D.G88] D.Geman. Random fields and inverse problems in imaging. In Springer Verlag, editor, *Lectures and Notes in Mathematics*, pages 113 – 193. Saint Flour lectures, 1988.
- [DM90] D.Williams and M.Shah. Edge contours using multiple scales. *Computer Vision Graphics Image Process*, 51:256 – 274, 1990.
- [DM92] D.Williams and M.Shah. A fast algorithm for active contours and curvatura estimation. *CVGIP:Image Understanding*, 55(1):14 – 26, 1992.
- [D.T86] D.Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(4):413 – 424, 1986.
- [D.T88] D.Terzopoulos. Deformable models. *The Visual Computer*, 4:306 – 331, 1988.
- [J.B74] J.Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Soc., Series B*, 39:192 – 236, 1974.
- [J.B86] J.Besag. On the statistical analysis of dirty pictures. *J. Royal Statistical Soc., Series B*, 48(3):855 – 867, 1986.
- [J.C86] J.Canny. A computational approach to edge detection. *IEEE Trans Patern Analysis and Machine Intelligence,PAMI-8*, pages 679 –689, 1986.
- [J.M74] J.Moussouris. Gibbs and markov systems with constraints. *Journal of statistical phisics*, 10:11 – 33, 1974.
- [KTY93] K.Yoshino, T.Kawashima, and Y.Aoki. Dynamic reconfiguration of active net structure. *Proc. Asian Conf. Computer Vision*, pages 159 – 162, 1993.
- [LI84] L.Cohen and I.Cohen. Finite elements methods for active contours models and ballons for 2d and 3d image. *IEEE Trans.Patern Analysis and Machine Intelligence*, 6(11):1131 – 1147, 1984.
- [MAD88] M.Kass, A.Witkin, and D.Terzopoulos. Snake: Active contours models. *Int.J.of Computer Vision*, 2:321 – 331, 1988.
- [M.B94] M.BroNielsen. Active net and cubes. *Submitted to CVGIP image Understanding*, 1994.
- [MK91] M.Sakaue and K.Yamamoto. Active net model and its application to region extraction. *JITE of Japan*, 45(10):1155 – 1163, 1991.

- [RR80] R.Kindermann and R.Snell. Markov random fields and their applications. *American Mathematical Society, providence*, 1980.
- [SD84] S.Geman and D.Geman. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Trans Pattern Analysis and Machine Intelligence*, 15(11):1131 – 1147, 1984.
- [YKK89] Y.Tsumiyama, K.Sakaue, and K.Yamamoto. Active net: Active net models for region extraction. *IPSJ SIG Notes, CV 63-2*, 89(96):1 – 8, 1989.