



UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS

ALGORITMO DE SEGMENTACIÓN DE OBJETOS BASADO
EN LA FORMA CON APLICACIÓN EN SISTEMAS DE
VIGILANCIA

L.M. ERBERTH JESÚS CASTILLO CEH

2015

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

Agradecimientos

A mi familia, especialmente a mis padres Erberth Castillo y Beatriz Ceh por el cariño, apoyo incondicional y confianza que depositaron en mí. Gracias por todo lo que me han otorgado para alcanzar y culminar esta etapa de mi vida. Sin ustedes nada de esto hubiera podido realizar.

A mi novia Sandra Pineda, que de igual forma me apoyo dándome ánimos y muchas veces escuchó mis ideas y conclusiones. Gracias por tu tiempo, cariño y apoyo.

A mis asesores y sinodales Dr. Carlos Brito y Dr. Víctor Uc, y de igual manera a mi sinodal Dr. Jorge Lugo, que dedicaron gran parte de su tiempo para la revisión y corrección de este trabajo. Les agradezco su tiempo, paciencia y consejos que fueron muy valiosos e importantes para el desarrollo y finalización de esta tesis.

A la Facultad de Matemáticas y al CONACYT por brindarme todos los recursos necesarios para la realización y culminación de mi carrera profesional.

A todos los profesores y amigos, que fueron fuente de conocimientos, aprendizaje, consejos y compañía invaluable durante mi estancia en el posgrado.

Índice general

v *Índice de figuras*

VI *Índice de tablas*

1 *1 Introducción*

1 1.1 *Motivación*

8 1.2 *Etapas de procesamiento de un sistema de vigilancia*

13 *2 Preliminares matemáticos*

13 2.1 *Transformaciones homogéneas*

16 2.2 *Diferencias finitas*

18 2.3 *Descenso del gradiente*

20 *3 Métodos de segmentación*

20 3.1 *Introducción*

21 3.2 *Segmentación basada en bordes*

22 3.2.1 *Técnica del histograma de grises.*

22 3.2.2 *Método basado en el gradiente.*

23 3.3 *Segmentación basada en regiones*

23 3.3.1 *Método de umbralización*

24 3.3.2 *Región de crecimiento*

24 3.3.3 *División y unión de regiones*

25	3.4	<i>Segmentación basada en teoría especial</i>
25	3.4.1	<i>Segmentación basada en redes neuronales</i>
26	3.4.2	<i>Segmentación basada en agrupación</i>
27	3.5	<i>Segmentación basada en modelos</i>
28	3.5.1	<i>Modelo de contornos activos o snakes</i>
29	3.5.2	<i>Modelo de conjuntos de nivel</i>
30	3.5.3	<i>Modelo Mumford-Shah</i>
31	3.5.4	<i>Modelo Chan-Vese</i>

33 **4 Método de Tsai et al.**

33	4.1	<i>Ecuación de alineación</i>
36	4.2	<i>Análisis de la implementación computacional</i>
36	4.2.1	<i>Error en las transformaciones</i>
38	4.2.2	<i>Importancia del orden en la minimización</i>
38	4.3	<i>Ecuación de la curva implícita</i>
38	4.3.1	<i>Mapas de distancia</i>
45	4.3.2	<i>Extracción de descriptores</i>
48	4.4	<i>Adaptación de Chan-Vese</i>

55 **5 Resultados y aportación**

55	5.1	<i>Introducción</i>
56	5.2	<i>Curva inicial</i>
59	5.3	<i>Función de reemplazo</i>
64	5.4	<i>Adaptación de Chan-Vese</i>

66 **6 Conclusiones**

68 **Bibliografía**

Índice de figuras

8	1.1	<i>Esquema general de procesamiento en sistemas de vigilancia.</i>
13	2.1	<i>Representación gráfica de la función $T : A \rightarrow B$.</i>
14	2.2	<i>Transformaciones afines sobre una imagen</i>
15	2.3	<i>Representación de la composición $T_2 \circ T_1$.</i>
18	2.4	<i>Diferentes resoluciones en una malla</i>
37	4.1	<i>Transformación de una imagen: (a) imagen original, (b) traslación en x y y, (c) escalamiento, (d) rotación, (e) escalamiento y rotación.</i>
37	4.2	<i>Transformación de una imagen y aplicación de un filtro promedio.</i>
39	4.3	<i>Función implícita $\phi = x^4 - 5x^2 + 4$</i>
40	4.4	<i>Función implícita $\phi = x^2 - 1$</i>
45	4.5	<i>Representación gráfica de un mapa de distancias.</i>
47	4.6	<i>Implementación de la ecuación de descripción implícita de forma.</i>
54	4.7	<i>Resultado de la implementación: (a) curva inicial, (b) curva después de 400 iteraciones.</i>
56	5.1	<i>Imágenes iniciales</i>
57	5.2	<i>SDF aplicado al conjunto de imágenes inicial</i>
61	5.3	<i>Ilustración de la función de reemplazo</i>
62	5.4	<i>Función ξ aplicada al conjunto de imágenes inicial</i>
65	5.5	<i>Falta de información en una imagen transformada</i>
65	5.6	<i>Adaptación Chan-Vese usando eigenformas calculadas con la función ϕ y con la función ξ</i>

Índice de tablas

21	3.1	<i>Clasificación de métodos de segmentación</i>
58	5.1	<i>Costo computacional del cálculo de eigenformas usando la función ϕ.</i>
63	5.2	<i>Costo computacional del cálculo de eigenformas usando la función ξ.</i>
64	5.3	<i>Comparación de costo computacional entre las funciones ϕ y ξ.</i>

Capítulo 1

Introducción

1.1. Motivación

Se puede definir el término *vigilancia* como el monitoreo del comportamiento, actividades u otra información, usualmente de las personas, con el propósito de influenciar, manejar, dirigir o protegerlas^[18]. La vigilancia se puede llevar a cabo, en particular, por medio de equipos electrónicos como pueden ser las cámaras de circuito cerrado de televisión (CCTV). Este tipo de sistemas consisten en una serie de cámaras localizadas en múltiples ubicaciones remotas y conectadas a un conjunto de monitores, a menudo colocados en una sola sala de control, a través de interruptores (matriz de video)^{[13][32]}.

Tanto en el documento de Inc^[13] como en el de Valera y Velastin^[32] se puede encontrar que muchos de los sistemas de CCTV utilizan técnicas analógicas para la distribución y el almacenamiento de imágenes. En el segundo se comenta además que las cámaras de CCTV convencionales generalmente utilizan un dispositivo de carga acoplada digital (CCD) para capturar imágenes. Entonces la imagen digital se convierte en una señal de vídeo compuesto analógica, que está conectado a la matriz de CCTV, monitores y equipos de grabación. En general, esta conexión es realizada a través de cables coaxiales y cables seriales.

En Inc^[13], Russo^[25] y Valera y Velastin^[32], se puede encontrar una descripción de la evolución que han tenido a lo largo del tiempo los sistemas de video vigilancia basados en cámaras de CCTV:

- **Primera generación.** La evolución, en términos de tecnología, de los sis-

temas de vigilancia que se encuentran basados en video comenzó con los sistemas CCTV que son totalmente analógicos, es decir, las cámaras se controlan y transmiten video en un formato analógico. Estas secuencias de video son obtenidas, intercambiadas y enviadas a pantallas de monitoreo por medio de tecnología de matrices de conmutación analógica, que también proporcionan la transmisión del video a grabadoras de vídeo (**VCR**) analógicas para fines de grabación.

Aunque en principio, estos dispositivos analógicos proporcionan capacidades básicas para la vigilancia y la grabación, también traen consigo varios inconvenientes en el sentido operativo:

- a) La grabación basada en VCR no facilita la grabación y reproducción del video de manera simultánea, sino que se necesitan componentes separados a fin de grabar video durante el proceso de investigación.
- b) El proceso de grabación es propenso a errores humanos: por ejemplo al momento de reemplazar las cintas grabadas por cintas en blanco o por el hecho de asegurarse que la grabación fue activada.
- c) Desde la perspectiva de la fiabilidad y disponibilidad del sistema, las fallas en el sistema de grabación pueden pasar desapercibidas si se realizan durante un período prolongado de tiempo.
- d) En cuanto al almacenamiento y la accesibilidad, los dispositivos producen cientos de cintas de video que luego deben ser vistas por los guardias de seguridad. Diversos estudios han demostrado que una persona que monitorea un video durante varias horas al día, resulta en un sistema de seguridad inefectivo ya que incluso después de 20 minutos de ver y analizar los videos, la atención de la mayoría de las personas ha degenerado hasta muy por debajo del nivel aceptable. Por otro lado, la visualización de vídeo en directo o grabado está limitado a operaciones específicas y centros de investigación. Para revisar el vídeo grabado desde un lugar remoto se requiere que la cinta apropiada sea localizada y enviada al centro de investigación. Además, las búsquedas manuales de estas cintas pueden tomar mucho tiempo para propor-

cionar información vital necesaria para ayudar en las investigaciones.

- e) En general se tiene el problema del costo de emplear personal de seguridad para supervisar cientos de cámaras.
- f) Las cintas de vídeo pueden tener mala calidad de imagen y se deterioran con el tiempo.
- g) Finalmente, de los sistemas de vídeo analógico no se puede extraer información de manera inteligente para el uso en los negocios.

- **Segunda generación.** La conversión de digital a analógico provoca cierta degradación de la imagen y la señal analógica es susceptible al ruido. Sin embargo, es posible tener sistemas digitales de CCTV aprovechando el formato digital inicial de las imágenes capturadas y mediante el uso de ordenadores de alto rendimiento. La mejora tecnológica proporcionada por estos sistemas ha llevado al desarrollo de sistemas semi-automáticos, conocidos como sistemas de vigilancia de segunda generación. La mayor parte de la investigación en sistemas de vigilancia de segunda generación se basa en la creación de algoritmos para la detección de eventos en tiempo real automáticos ayudando al usuario a reconocer los hechos.

Los sistemas de esta generación también se basan, como dispositivo de entrada, en la cámara analógica, conectividad coaxial o por fibra y conmutación de vídeo proporcionada por un conmutador de matriz de vídeo analógico. Sin embargo, las funciones de grabación se han mejorado, pues estos sistemas se centran principalmente en abordar los problemas de grabación y almacenamiento y para este propósito, las VCR analógicas son reemplazadas por grabadoras de video digital (**DVR**). Las DVR convierten el video analógico de entrada a un formato digital y el video resultante es almacenado en unidades de disco duro internas o en el almacenamiento local de conexión directa. Adicionalmente, las bases de datos internas de las VCR reducen el tiempo de recuperación de video durante las investigaciones.

Como parte de las ventajas o novedades de esta generación, los DVR pueden ayudar a las organizaciones a lograr un mejor rendimiento de sus inversiones en seguridad a través de:

- a) Activación de la detección en tiempo real y la prevención potencial de incidentes de seguridad mediante una mejor recopilación de inteligencia.
- b) El uso de la visión basada en eventos con fines de investigación, lo que elimina la necesidad de revisar cronológicamente videocintas.
- c) Reducción de la necesidad de controlar las cámaras de vídeo y cambiar las cintas.
- d) Aumento de la seguridad del producto, al disuadir a los ladrones potenciales y seguimiento de personal.
- e) Proporcionar pruebas contra reclamaciones fraudulentas.
- f) El aumento de la seguridad interior al igual que en los estacionamientos.

Por otro lado, como desventaja se puede mencionar que algunos DVR utilizan sistemas operativos de computadoras personales que pueden estar sujetos a la manipulación y el virus de propagación, por lo que los DVR deben incluirse en un programa de mantenimiento preventivo para la protección contra virus regulares y para la configuración de mecanismos de seguridad. Además, con frecuencia el software de DVR es accedido y controlado por una interfaz de usuario específica del proveedor a menudo ejecutándose como un conjunto de aplicaciones de administrador y operador en una computadora personal (PC). Como tal, los DVR de segunda generación requieren con frecuencia un cliente de visualización de PC. El uso de software cliente ofrece unas concesiones; puede limitar el acceso a vídeo grabado sobre una base local, que puede ser deseable, pero también puede imponer problemas en situaciones de emergencia donde la visión remota, sobre una red IP puede ser útil.

Finalmente algunos DVR se puede acceder a través de un PC conectado a la red para reducir aún más el tiempo asociado con el archivado y la recuperación de video. El acceso a petición de video archivado acelera la revisión de la evidencia y mejora el control de las pruebas. También ahorra tiempo y

esfuerzo; los investigadores no tienen que viajar a otros centros para llevar a cabo las investigaciones.

- **Tercera generación.** La vigilancia inteligente, la vigilancia de video inteligente, análisis de video, de video inteligente y análisis inteligentes son nombres típicos usados para describir el concepto de la aplicación de análisis de la señal automatizada y reconocimiento de patrones para cámaras de video y sensores, con el objetivo de extraer de forma automática “información útil” de flujos de video y sensores.

Por su parte, el término sistemas de vigilancia de tercera generación es usado a veces en la literatura para referirse a los sistemas concebidos para hacer frente a un gran número de cámaras, distribución geográfica de los recursos, muchos puntos de control, y para reflejar la naturaleza jerárquica y distribuida del proceso humano de la vigilancia. Es importante destacar que esos son requisitos previos importantes, si tales sistemas se van a integrar como parte de una herramienta de gestión.

Los principales objetivos que se esperan de una aplicación de vigilancia de tercera generación, basado en los requisitos de los usuarios finales, son: proporcionar una buena comprensión de la escena, orientada a atraer la atención del operador humano en el tiempo real, posiblemente, en un entorno multi-sensor; proporcionar información de la vigilancia y el uso de componentes estándar de bajo costo.

Otro aspecto relevante es la seguridad de las comunicaciones entre los módulos, pues los datos pueden necesitar ser enviados a través de redes abiertas y hay temas críticos en el mantenimiento de la privacidad y la autenticación de datos.

Al igual que con los despliegues de video vigilancia de primera y segunda generación, los despliegues de tercera generación se basan principalmente en la cámara analógica como entrada, conectividad por fibra o coaxial y conmutación de vídeo proporcionado por una matriz de conmutación de vídeo analógico. Sin embargo, la accesibilidad de vídeo en directo y grabado se ve reforzada.

- **Cuarta generación.** La videovigilancia de cuarta generación proporciona beneficios adicionales y ventajas con respecto a las generaciones anteriores: se expande y amplía la capacidad de las puertas de enlace de video vigilancia (mejorados codificadores y decodificadores) y el NVR, lo que permite que el conmutador de matriz sea sustituido por plataformas de conmutación Ethernet estándar y típicas de menor costo.

humano en el tiempo real, posiblemente, en un entorno multi-sensor, información de la vigilancia y el uso de componentes de bajo costo estándar. grado de interoperabilidad del vendedor; operadores pueden utilizar su teclado preferido y el joystick de un solo proveedor, mientras que la visualización de vídeo con cámaras de otro fabricante. Puertas de enlace IP de Cisco proporcionan -cualquier- a cualquier proveedor de interoperabilidad, y proteger las inversiones en cámaras de video vigilancia analógicas y estaciones de monitoreo. por ciento. una "alarma de puerta forzada" fue un golpe de viento, y no un intruso.

Por otro lado, Devasena et al.^[7] y Valera y Velastin^[32] concuerdan en que la creciente demanda de seguridad por parte de la sociedad conduce a una necesidad también creciente de las actividades de vigilancia en muchos entornos. En particular, la demanda de monitorización remota para propósitos de seguridad ha sido objeto de especial atención. Con todo esto, la video vigilancia ha incrementando significativamente su acercamiento a las organizaciones que buscan seguridad o resguardo de sus bienes. Al mismo tiempo, la necesidad de observar más personas, lugares o cosas, junto con el deseo de obtener más información útil a partir de datos de un video, está motivando nuevas demandas de escalabilidad y capacidad entre otras características. En otras palabras, esto se ha traducido en el despliegue de sistemas de CCTV de gran tamaño.

Para ejemplificar lo mencionado con anterioridad Hu et al.^[11] presenta algunos escenarios en los que resulta de utilidad la vigilancia:

- En los escenarios públicos, se puede ayudar a la policía para atrapar sospechosos instalando los sistemas en lugares donde normalmente aparecen estas personas. De igual forma pueden monitorear autopistas y analizar el flujo del tráfico o la congestión en las carreteras que son de gran importancia.

- En áreas especiales como bases militares o unidades gubernamentales importantes, solo personas autorizadas pueden tener acceso, por lo cual cuando una persona desee entrar, el sistema podría obtener automáticamente las características del visitante como puede ser, la altura, el aspecto facial, el modo de caminar, entre otras características, a partir de imágenes tomadas en tiempo real, y luego decidir si el visitante puede tener acceso.
- En los escenarios comerciales, es importante determinar cuándo ciertos comportamientos de las personas o vehículos son normales o anormales, esto debido a que si el comportamiento se detecta como anormal, puede ser un indicativo de robo.

Por todo este interés en la vigilancia como por ejemplo en los escenarios anteriores, entre otros, ha aumentando la necesidad de crear e implementar sistemas de vigilancia inteligentes o automatizados. Estos sistemas de vigilancia inteligentes, tratan con los monitoreos en tiempo real de objetos persistentes y transitorios de un entorno específico. Los principales objetivos de esos sistemas son proveer una interpretación automática de escenas y entender y predecir las acciones e interacciones de los objetos observados basado en la información adquirida por los sensores.

Para poder hacer frente a toda esa creciente demanda, se ha llevado a cabo investigación y desarrollo de forma continua, entre otros, en entornos comerciales y académicas para encontrar mejoras o nuevas soluciones de procesamiento de señales, comunicaciones, ingeniería de sistemas y la visión por computador.

Los sistemas de vigilancia creados con fines comerciales tienden a usar el hardware de propósito específico y un uso creciente de redes de cámaras inteligentes digitales. Las tareas de procesamiento comunes que realizan estos sistemas son la intrusión y de detección de movimiento y la detección de paquetes.

Por otro lado, la investigación en el mundo académico tiende a mejorar las tareas de procesamiento de imágenes mediante la generación de algoritmos más precisos y robustos en la detección de objetos y el reconocimiento, seguimiento, reconocimiento de la actividad humana, base de datos y herramientas de evaluación del desempeño de seguimiento.

1.2. Etapas de procesamiento de un sistema de vigilancia

Dado como entrada un conjunto o secuencia de imágenes, o más aún un video en general, las principales etapas de procesamiento en un sistema de vigilancia inteligente son: detección o segmentación de objetos en movimiento, reconocimiento o clasificación de esos objetos, seguimiento y análisis del comportamiento como se ilustra en la *Figura 1.1*.

La detección de objetos o segmentación en tiempo real de las regiones en movimiento es un paso elemental en muchos de estos sistemas^[7], ya que sólo estas regiones son necesarias considerar para los procesos posteriores^[11].

Los autores Devasena et al.^[7], Hu et al.^[11] y Valera y Velastin^[32] coinciden en que existen principalmente dos enfoques convencionales para esta etapa de procesamiento: “diferencia temporal” y “sustracción de fondo”. El primer enfoque consiste en la sustracción de dos, o en el caso de una versión mejorada, tres tramas o secuencias de imágenes consecutivas seguidas de la aplicación de una función de umbralización¹ que es utilizada para detectar cambios. Esta técnica tiene un buen rendimiento en entornos dinámicos, ya que es muy adaptable, pero tiene un bajo rendimiento en la extracción de todos los objetos en los píxeles correspondientes.

La segunda técnica se basa en la diferencia pixel a pixel entre un modelo o fondo de la imagen de referencia y la imagen actual, seguido por un proceso de etiquetado. La sustracción de fondo tiene un mejor rendimiento en la extracción de información de los objetos, pero es sensible a los cambios dinámicos en el medio ambiente como puede ser un cambio en la iluminación. Después de la aplicación

¹Función mediante la cual se obtiene una imagen con sólo dos niveles de color. Usualmente aplicado para separar objetos y fondo.

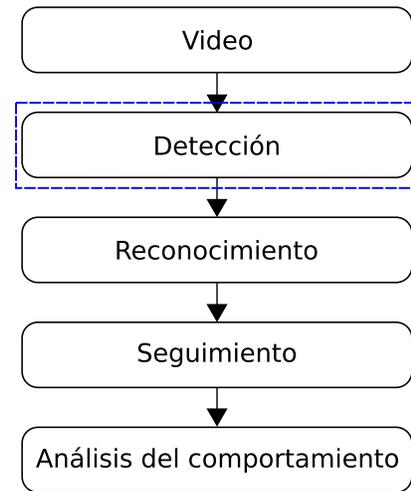


Figura 1.1: Esquema general de procesamiento en sistemas de vigilancia.

de uno de estos dos enfoques, típicamente se aplican operaciones morfológicas² para reducir el ruido de la diferencia de imágenes.

Otras técnicas para el modelado de fondo para la detección de objetos, son por ejemplo el método de sustracción de fondo adaptativo, el método de mezcla Gaussiana adaptativa, métodos de seguimiento del cuerpo humano 2-D y 3-D, entre otros. Además, se pueden realizar más etapas de pre-procesamiento para mejorar la calidad de las imágenes, ya que debido a la cámara, la iluminación, los reflejos, entre otras variables, los fotogramas de video pueden contener una gran cantidad de ruido. Esto significa una gran cantidad de píxeles cuyos valores no corresponden al valor de la imagen real.

La siguiente etapa de procesamiento puede considerarse como un problema estándar de reconocimiento de patrones, debido a que diferentes regiones móviles, pueden corresponder a diferentes objetos en movimiento en ambientes naturales. Por ejemplo las escenas de tráfico probablemente incluyen seres humanos, vehículos, aves, nubes en movimiento, entre otros objetos, y desde luego es esencial clasificar correctamente los objetos de interés para hacer un mejor seguimiento y análisis de comportamiento^[11]. Principalmente, existen dos categorías de enfoques para la clasificación de objetos en movimiento: clasificación basada en la forma y clasificación basada en el movimiento.

Para el tercer punto, el seguimiento de objetos consiste en la detección de objetos de interés en movimiento y trazar su recorrido por el análisis de los mismos. Entonces como se puede observar, cualquier método de seguimiento requiere un mecanismo de detección de objetos en cada frame o en la primera aparición del objeto en el vídeo.

Un enfoque común para la detección de objetos es utilizar la información en un solo cuadro. Pero, algunos métodos de detección de objeto utilizan la información cronológica calculada a partir de una secuencia de frames para reducir el número de falsas detecciones. Esta información temporal se encuentra por lo general en forma de diferencia de frames, que pone de relieve las regiones cambiantes en frames consecutivos. Teniendo en cuenta las regiones del objeto en la imagen, es entonces la tarea del rastreador realizar la correspondencia del objeto de un frame

²Operaciones aplicadas a las imágenes que las simplifican y conservan sus principales características de forma de sus objetos.

al siguiente para generar las pistas.

Continuando con esta etapa de procesamiento, las técnicas de seguimiento se pueden dividir principalmente en dos enfoques: los modelos 2-D que pueden o no contar con modelos de forma explícita y los modelos 3-D. El enfoque basado en el modelo utiliza un explícito conocimiento geométrico a priori de los objetos a seguir, lo que en aplicaciones de vigilancia por lo general son las personas, vehículos o ambos. Este conocimiento a priori se puede conseguir mediante el cálculo de la apariencia del objeto en función de su posición relativa a la cámara. La geometría de la escena se obtiene de la misma manera. Con el fin de construir modelos de la forma, el uso de técnicas de calibración de la cámara se vuelve importante.

Una vez que el conocimiento a priori está disponible, puede ser utilizado en un algoritmo de seguimiento robusto frente a condiciones variables tales como el cambio de la iluminación, que ofrece un mejor rendimiento en la solución de (auto) oclusiones o (auto) colisiones.

Un método de seguimiento común es utilizar un mecanismo de filtrado para predecir cada movimiento del objeto reconocido. El filtro más utilizado en los sistemas de vigilancia es el filtro de Kalman. Adecuadas cajas o elipses delimitadores, lo que comúnmente se denominan "notas", a las regiones de imagen de máxima probabilidad es otro método de seguimiento basado en modelos estadísticos.

En algunas situaciones de interés las hipótesis formuladas para aplicar filtros Gaussianos o lineal no se sostienen, y entonces se han propuesto filtros bayesianos no lineales, tales como filtros de Kalman extendido (EKF) o filtros de partículas. Otras técnicas de sustracción de fondo utilizados para el seguimiento de objetos en la vigilancia de vídeo que van desde métodos sencillos, utilizados para maximizar la velocidad y frenar los requisitos de memoria, hasta los enfoques más complicados, que se utiliza para lograr la máxima precisión posible en cualquier circunstancia potenciales, todos destinados a la ejecución en tiempo real, son: media de Gauss corregido, filtro de mediana temporal, mezcla de gaussianas, estimación de densidad de kernel (KDE), aproximación KD secuencial, co-ocurrencia de variaciones de imágenes y la técnica de eigen fondos.

Entre esos últimos métodos, los métodos simples, como el promedio de Gauss o el filtro de mediana ofrecen una precisión aceptable mientras logran una alta velocidad de cuadro y tienen los requisitos de memoria limitados. Los métodos como la mezcla de gaussianas y KDE demuestran una precisión muy buena de modelo. KDE tiene un alto requerimiento de memoria (en el orden de unos 100 frames) que impide la fácil implementación en dispositivos con poca memoria. SKDA es una aproximación de KDE que se presenta casi tan precisa, pero mitiga la necesidad de memoria en un orden de magnitud y tiene menor complejidad de tiempo. Métodos como el co-ocurrencia de las variaciones de las imágenes y los eigen fondos abordan explícitamente la correlación espacial. Ambos ofrecen una buena precisión contra el tiempo razonable y la complejidad de la memoria.

La última etapa del esquema general presentado, es el análisis o comprensión del comportamiento, que se puede pensar como la clasificación de los datos de características que varían a lo largo del tiempo, es decir, hacer coincidir un secuencia de prueba desconocida, con un grupo de secuencia de referencia etiquetadas que representan comportamiento típico. Entonces el problema fundamental para esta etapa de procesamiento, es aprender las secuencias de comportamiento de referencia a partir de muestras de entrenamiento, y diseñar tanto el entrenamiento como los métodos de comparación que afronten efectivamente pequeñas variaciones de los datos de características dentro de cada clase de patrones de movimiento^[11].

Cabe mencionar que aunque en el esquema general de procesamiento no se contempla, una de las etapas finales de un sistema de vigilancia es el almacenamiento y recuperación. Relativamente poca investigación se ha hecho en la forma de almacenar y recuperar toda la información de vigilancia obtenida de una manera eficiente, especialmente cuando es posible tener diferentes formatos de datos y tipos de información para recuperar. Sin embargo, para resolver los problemas como espacio de almacenamiento y consumo de tiempo para grabar y ver el video, Huang y Chen^[12] propusieron un método para registrar video únicamente cuando este contiene información importante, por ejemplo cuando en la escena se detecta movimiento.

Se puede concluir, como describen en su trabajo Hu et al.^[11], que el objetivo

de los sistemas de vigilancia inteligentes no es sólo poner las cámaras en lugar de los ojos humanos, sino también llevar a cabo tanto como sea posible toda la tarea de vigilancia a procesos automáticos. Todo esto es debido a que la videovigilancia pasiva tradicional ha demostrado volverse ineficaz cuando el número de cámaras es superior a la capacidad de los operadores humanos para supervisarlas, es decir, la tarea de vigilancia se torna más difícil mientras más cámaras existan. Debido a toda esta situación, se puede pensar en proporcionar ayudas visuales a las personas que manipulan los sistemas descritos anteriormente a través de algoritmos computacionales.

Para ilustrar la idea anterior, imaginando un escenario en el supermercado, estas ayudas pudieran indicar cuándo un objeto de interés, como puede ser mercancía en un carrito de compras, puede convertirse en un objeto de robo tanto si al comprador se le olvida extraer la mercancía del carrito como si intencionalmente no permite su cobro. Una forma de realizar esto puede ser segmentar los objetos presentes en una escena de video, realizar un proceso de clasificación para conocer de que tipo de objeto se trata (objeto de interés o no) y emitir una alerta al operador del sistema de vigilancia.

Finalmente, como se describió al principio de esta sección, una parte fundamental de la solución esbozada es la segmentación del objeto, como queda ilustrada en la *Figura 1.1*. Desde luego, si la segmentación es muy mala, la clasificación y en general los procesos posteriores fallarán. Por lo tanto esta etapa de procesamiento es el objeto de interés en este documento.

Capítulo 2

Preeliminarios matemáticos

2.1. Transformaciones homogéneas

En general, dados dos conjuntos A y B , los cuales son llamados dominio e imagen respectivamente, se define una función T como una regla que asocia a cada elemento $a \in A$, un elemento de B que es denotada como $T(a)$.

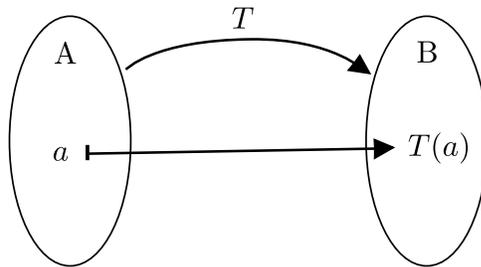


Figura 2.1: Representación gráfica de la función $T : A \rightarrow B$.

Sea T una función definida como $T : A \rightarrow B$, luego se establecen las siguientes definiciones:

- T es una función inyectiva, si se cumple que $T(a_i) = T(a_j)$ implica que $a_i = a_j$. Es decir, si las imágenes de dos elementos del dominio son iguales, entonces tales elementos del dominio son iguales. En otras palabras, a cada elemento distinto en el dominio, le corresponde un elemento distinto en la imagen.
- T es una función suprayectiva o sobre, si se cumple que para cada $b \in B$,

existe un elemento $a \in A$ tal que $T(a) = b$. Es decir, cada elemento del conjunto imagen tiene asociado un elemento en el conjunto dominio. En otras palabras, T es aplicada a todo el conjunto imagen.

- T es una función biyectiva, si se cumple que es tanto inyectiva como suprayectiva.

En un caso particular, sea T una transformación¹ definida en $X \subseteq \mathbb{R}^2$ y $Y \subseteq \mathbb{R}^2$ como:

$$T : X \longrightarrow Y \quad (2.1)$$

Más aún, si la transformación T preserva líneas paralelas o equivalentemente preserva colinealidad, entonces se dice que T es una *transformación afín*.

En esta sección, se presentan tres transformaciones afines definidas como la ecuación 2.1 (en cuanto al dominio y la imagen), las cuales son: traslación, rotación y escalamiento. La *Figura 2.2* ilustra el efecto de cada transformación a un objeto dado.

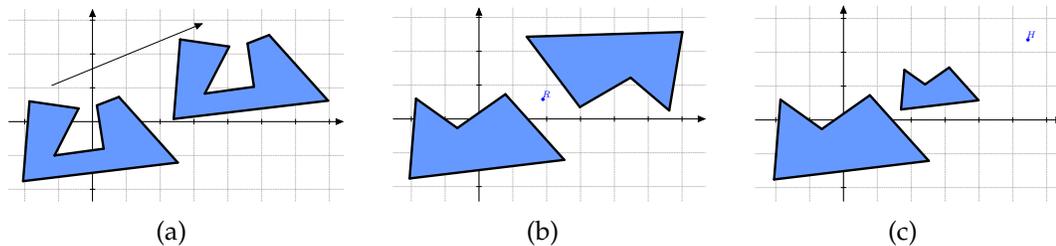


Figura 2.2: Transformaciones afines: (a) traslación, (b) rotación, (c) escalamiento.

Cada una de esas transformaciones tiene asociada una representación como matriz. Sin embargo, para esta representación se utilizan las *coordenadas homogéneas*, lo que consiste en representar los vectores $x \in \mathbb{R}^2$ como vectores en \mathbb{R}^3 incluyendo el neutro multiplicativo en la tercera entrada del vector:

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightsquigarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

¹Utilizaremos el término *transformación* para referirnos a una función biyectiva

Con estas modificaciones, las matrices para la traslación, escalamiento y rotación, son respectivamente:

$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} h & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Sea ahora T_1 una transformación definida en $A \subseteq \mathbb{R}^2$ y $B \subseteq \mathbb{R}^2$, y sea T_2 una transformación definida en B y $C \subseteq \mathbb{R}^2$ como sigue:

$$T_1 : A \longrightarrow B \quad \text{y} \quad T_2 : B \longrightarrow C$$

entonces se define la composición de transformaciones, denotada $T_2 \circ T_1$, como

$$(T_2 \circ T_1)(x) = T_2(T_1(x)), \quad x \in A$$

En decir, dado un elemento $x \in A$, la composición se obtiene al aplicar T_1 a x y seguidamente aplicar T_2 al resultado obtenido, en otras palabras aplicar T_2 a la imagen $T_1(x)$ de x .

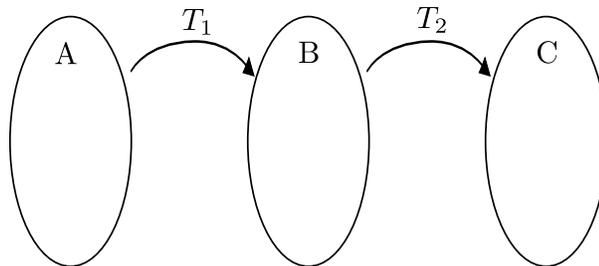


Figura 2.3: Representación de la composición $T_2 \circ T_1$.

Finalmente, cabe mencionar que la operación análoga para la representación matricial será el producto de matrices. Y se debe notar que dado que la multiplicación de matrices no es conmutativa, la correspondiente operación de composición tampoco lo es.

El lector interesado en profundizar más sobre este y otros temas relacionados, puede encontrar más información en los libros de Gentle^[8], Kelly y Weiss^[17],

Rees^[24] y Snapper y Troyer^[28].

2.2. Diferencias finitas

Supongamos que se desea calcular la derivada de una función continua respecto a una variable específica. En general, una *derivada parcial* respecto a la variable x_i , se obtiene mediante el límite

$$\lim_{h \rightarrow 0} \frac{f(a_1, \dots, a_i + h, \dots, a_n) - f(a_1, \dots, a_n)}{h}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y $a \in \mathbb{R}^n$.

En particular, para $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, las derivadas respecto a las componentes x y y son respectivamente:

$$\lim_{h \rightarrow 0} \frac{f(a_1 + h, a_2) - f(a_1, a_2)}{h} \quad \text{y} \quad \lim_{h \rightarrow 0} \frac{f(a_1, a_2 + h) - f(a_1, a_2)}{h} \quad (2.2)$$

En análisis numérico, una forma para aproximar estas derivadas es utilizar el método de *diferencias finitas* que consiste en aproximaciones locales a la derivada parcial. Este método reemplaza un operador diferencial continuo con una aproximación de diferencias.

Basándose en las expresiones de la ecuación 2.2, la aproximación de la primera derivada de una función f evaluada en el vector $\mathbf{x} = (x_1, x_2)$, por ejemplo para la primera componente, es mediante la fórmula:

$$f'(\mathbf{x}) \approx \frac{f(x_1 + h, x_2) - f(x_1, x_2)}{h} \quad (2.3)$$

Esta aproximación suele ser llamada *derivada hacia adelante*. Por otro lado, la aproximación

$$f'(\mathbf{x}) \approx \frac{f(x_1, x_2) - f(x_1 - h, x_2)}{h} \quad (2.4)$$

es conocida como *derivada hacia atrás*.

Realizando un promedio de las ecuaciones 2.3 y 2.4, se obtiene la ecuación

$$f'(\mathbf{x}) \approx \frac{f(x_1 + h, x_2) - f(x_1 - h, x_2)}{2h}$$

conocida como *derivada central*, la cual presenta una mejor aproximación a la derivada parcial respecto a la primera componente. En general, esta derivada tiene una mejor aproximación que las otras diferencias finitas.

Además de contar con la aproximación de las derivadas anteriores, otro paso importante es la discretización del dominio que se realiza definiendo una cuadrícula (a partir de ahora, malla) sobre ese dominio. Entonces utilizando el método de las diferencias finitas, una implicación directa es que el valor de h en esas ecuaciones, dependerá de la resolución a la que se defina la malla que se utilice para representar la superficie.

De manera más formal, una malla sobre un intervalo $[a, b]$, es una partición de ese intervalo por el conjunto de puntos $P = \{x_1, \dots, x_n\}$, de tal forma que se cumplen las siguientes relaciones:

$$a = x_1 < x_2 < \dots < x_{n-1} < x_n = b$$

Luego, una malla sobre una región $[a, b] \times [c, d]$, no es más que la operación

$$M_{[a,b] \times [c,d]} = M_{[a,b]} \times M_{[c,d]}$$

donde $M_{[a,b]}$ y $M_{[c,d]}$ son las mallas para los intervalos $[a, b]$ y $[c, d]$ respectivamente.

Con la definición anterior, podemos definir el valor de h como la distancia entre dos puntos de P , es decir, $h = x_j - x_i$, con $i < j$. En otras palabras h será el espacio entre los puntos que generan la partición del intervalo, suponiendo una partición uniforme.

Esto significa que a medida que se tenga una malla más fina, como la de la *Figura 2.4(b)* comparada con la malla de la *Figura 2.4(a)*, mejor será la aproximación a la derivada parcial, ya que h tenderá más hacia el valor cero. La afirmación anterior se cumplirá debido a que definir una malla más fina, se traduce en tener menor distancia entre los puntos que generan la partición del dominio, o

en otras palabras, más elementos en el conjunto P de partición.

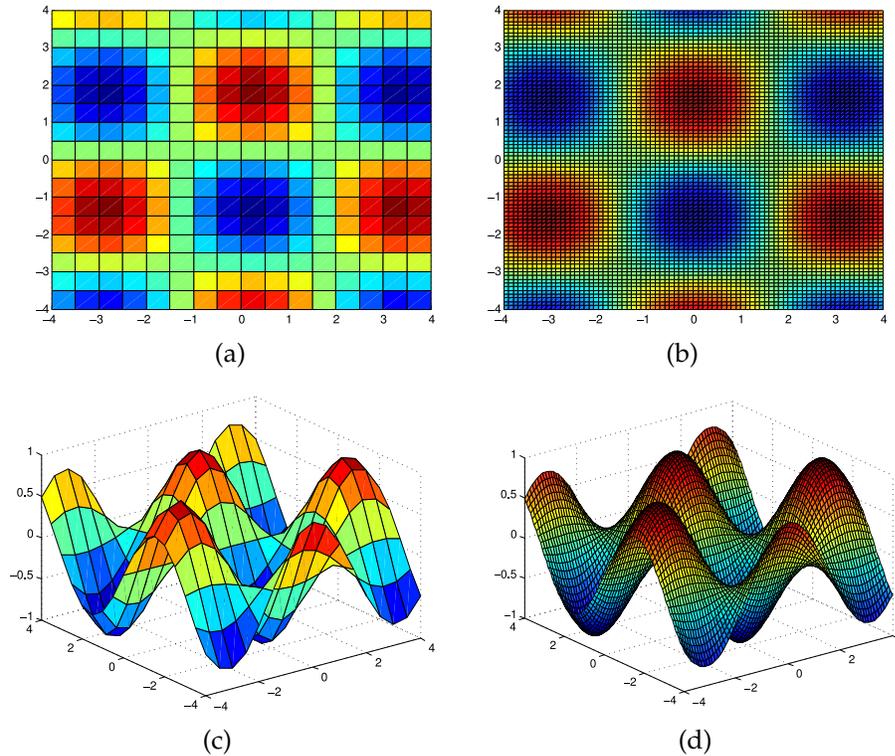


Figura 2.4: Diferentes resoluciones en una malla para representar la función $f(x,y) = \sin(x)\cos(y)$: las figuras (a) y (c), son la representación de f con una malla de 16×16 particiones, las figuras (b) y (d), son la representación de f con una malla de 80×80 particiones.

Para mayor aclaración sobre este y otros temas relacionados, el lector interesado puede consultar la información en Osher y Fedkiw^[22], Spivak^[29] y Thomas^[30].

2.3. Descenso del gradiente

Sea $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ dada por la expresión

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b$$

donde $b \in \mathbb{R}^n$ y $A \in \mathbb{R}^{n \times n}$ se asume que es definida positiva y simétrica. Minimizar ϕ y resolver el sistema $Ax = b$ son problemas equivalentes si A cumple las

restricciones anteriores^[9].

Recordemos que el gradiente de una función ϕ evaluada en un punto \mathbf{x} , denotado como $\nabla\phi(\mathbf{x})$, es una función vectorial cuya i -ésima función componente es la derivada parcial de ϕ con respecto a x_i , denotado como

$$\frac{\partial\phi(\mathbf{x})}{\partial x_i}$$

De cálculo se sabe que en un punto \mathbf{x} donde el vector gradiente es diferente de cero, el gradiente negativo, $-\nabla\phi(\mathbf{x})$, señala hacia los valores más bajos de la función. En efecto, $-\nabla\phi(\mathbf{x})$, es la dirección local de una de las más simples estrategias para minimizar ϕ , el cual es llamado el *método del descenso del gradiente*, en el sentido de que los valores de la función decrecen más rápidamente a lo largo de la dirección del gradiente negativo que a lo largo de cualquier otra dirección^{[9][23]}.

Comenzando desde un punto inicial x_0 , cada solución aproximada sucesiva esta dada por la expresión

$$x_{n+1} = x_n - \alpha_n \nabla\phi(x_n)$$

donde α_n es un parámetro de línea de búsqueda que determina qué tan lejos se avanza en esa dirección.

A pesar de que este método es bastante fiable por su capacidad de avanzar siempre y cuando el gradiente sea distinto de cero, las iteraciones resultantes pueden avanzar de ida y vuelta, en zig zag. Esto se debe a que el nuevo gradiente en el mínimo punto de una línea de minimización es perpendicular a la dirección atravesada, esto es, se realizan giros en ángulos rectos, los cuales no necesariamente llevan al mínimo. Por estas razones este método hace muy lento el proceso de búsqueda de soluciones^{[9][10]}.

Capítulo 3

Métodos de segmentación

3.1. Introducción

La segmentación de imágenes es un problema fundamental en el análisis y procesamiento de imágenes y visión computacional^[14]. Consiste en el proceso de particionar una imagen digital $f(x, y)$ en regiones, clases o subconjuntos continuos, es decir conjuntos de píxeles, que son disconexos y no vacíos f_1, f_2, \dots, f_n ^[15], y que son similares respecto a un criterio de homogeneidad tal como el color, la intensidad o textura, con el fin de localizar e identificar objetos y sus fronteras en la imagen, lo que permite una mejor extracción de atributos^{[6][26]}.

Se pueden encontrar cientos de técnicas de segmentación en la literatura, pero no existe un solo método que pueda considerarse bueno para imágenes diferentes. Por tanto, uno de los temas complicados ha sido desarrollar una estrategia unificada para la segmentación de imágenes el cual sea aplicado a todos los diferentes tipos de imágenes, incluso la selección de una técnica para una tipo de imagen específico es un problema difícil.

En general, la clasificación de las técnicas de segmentación de imágenes están basados en dos propiedades de la imagen^{[6][15]}:

- **Discontinuidad:** las imágenes pueden ser divididas basándose en los cambios abruptos de intensidades.
- **Similitud:** las separación de la imagen se lleva a cabo, basándose en regiones que son similares de acuerdo a un conjunto de criterios predefinidos.

Basadas en las dos propiedades anteriormente mencionadas, una de las clasificaciones de las diferentes técnicas de segmentación de imágenes, recopiladas según Dass y Devi^[6], Jiang et al.^[14], Kang et al.^[15] y Selvaraj Assley y Chellakkon^[26], puede considerarse como se muestra en el siguiente esquema

Categorías principales	Subclases
Segmentación basada en bordes	Técnica del histograma de grises
	Método basado en gradiente
Segmentación basada en regiones	Método de umbralización
	Región de crecimiento
	División y unión de regiones
Segmentación basada en teoría especial	Segmentación basada en redes neuronales
	Segmentación basada en agrupación
Segmentación basada en modelos	Modelo de contornos activos o snakes
	Modelo de conjuntos de nivel
	Modelo Mumford-Shah
	Modelo Chan-Vese

Tabla 3.1: Clasificación de algunos métodos de segmentación de imágenes.

Cabe mencionar que existen distintas clasificaciones según el autor y existen muchos métodos más, pero en este apartado solamente se mencionan aquellos que en general se consideran más conocidos o utilizados.

3.2. Segmentación basada en bordes

Un borde es un conjunto de píxeles relacionados entre sí, que se encuentran sobre una frontera entre regiones diferentes, donde existen intensas discontinuidades como pueden ser cambios de gris, diferencias de color, variedad de texturas, entre otros.

El resultado de la implementación de esta técnica, es una imagen binaria y para obtener un resultado satisfactorio se debe mantener un equilibrio entre la precisión de la detección y la inmunidad al ruido. Si el nivel de precisión es alto, el ruido puede conducir a falsos bordes haciendo el contorno de las imágenes irrazonable. Por otro lado, algunas partes del contorno pueden no ser detectadas y la posición del objeto puede ser equivocada si la inmunidad al ruido es excesiva.

Basado en esta teoría, existen dos principales métodos de segmentación: el método del histograma de grises y el método basado en el gradiente. Además, también los operadores Laplacianos pueden ser utilizados para la detección de los bordes.

Como ventajas se puede mencionar que los algoritmos basadas en este enfoque son computacionalmente rápidos y no necesitan información a priori acerca del contenido de la imagen. Por el contrario, entre las desventajas encontramos que los métodos basados en gradiente o Laplacianos son sensibles al ruido y solamente trabajan adecuadamente cuando las imágenes tienen un buen contraste entre diferentes regiones. Además, este enfoque detecta todos los bordes y es muy difícil encontrar la relación entre bordes y regiones de interés.

3.2.1. Técnica del histograma de grises.

En la implementación de esta técnica, el resultado depende principalmente de la selección del umbral T . Encontrar la intensidad máxima y mínima de nivel de gris es difícil por que el histograma es desigual debido al ruido, así que se sustituye aproximadamente las curvas correspondiente al objeto y al fondo, por dos curvas gaussianas cónicas, cuya intersección es el valle del histograma. Por último, el umbral buscado T es el valor de gris en el punto de intersección de ese valle.

3.2.2. Método basado en el gradiente.

Este método, funciona bien cuando el cambio de valores en el nivel de gris, es suficientemente intenso cerca del borde y existe poco ruido en la imagen. Si la magnitud del gradiente presenta valores altos, esto es indicador posible de rápida transformación entre dos diferentes regiones.

Los operadores de detección de bordes más comúnmente utilizados son el operador Sobel, operados Canny, operador de Laplace, operador Laplaciano del Gaussiano, entre otros. Entre ellos, la técnica de Canny es el más prometedor, aunque toma más tiempo en comparación con el operador Sobel. Canny propuso tres criterios para la detección de bordes: resultado de la detección óptima, re-

sultado de la posición óptima y baja repetición de respuesta. Basado en esos tres criterios, inventó el “filtro lineal óptimo”, el cual es la primera derivada de la función Gaussiana.

3.3. Segmentación basada en regiones

Mientras que la segmentación basada en bordes particiona la imagen basándose en los cambios abruptos de intensidades de gris cerca de los bordes, la segmentación basada en regiones particiona la imagen en regiones similares de píxeles conectados de acuerdo a un criterio predefinido.

Comparados con los métodos basados en bordes, este tipo de técnicas de segmentación de imágenes son relativamente sencillas y más inmunes al ruido. Entre los métodos que incluyen este tipo de algoritmos encontramos el método de umbralización, región de crecimiento y el método de división y fusión de regiones.

3.3.1. Método de umbralización

Mediante esta técnica, el objetivo es agrupar los píxeles de una imagen en varias clases para separar objetos del fondo, basándose en un valor de nivel de gris específico llamado umbral. Cuando existe un sólo umbral T , la umbralización puede definirse como

$$f(x, y) = \begin{cases} 1, & \text{si } f(x, y) \geq T \\ 0, & \text{si } f(x, y) < T \end{cases}$$

por tanto, cualquier punto (x, y) para el cual $f(x, y) = 1$, es llamado punto del objeto, en caso contrario es llamado fondo.

En general, este método puede dividirse en umbralización global, si el umbral es seleccionado de la imagen completa, y umbralización local, cuando la imagen completa es dividida en varias subregiones y a cada una de ellas es asignada diferentes umbrales.

Para la selección del umbral, se puede hacer uso de la selección manual, que

se basa en el ensayo y error. Un segundo método puede realizarse el método basado en el histograma y la media o la mediana. Finalmente, el tercer método es un algoritmo de selección de umbral unimodal que no requiere mucho conocimiento específico de la imagen. La limitación en umbralización automática es que la complejidad incrementa con el tamaño de la imagen, dado que la media se calcula en cada iteración. Sin embargo, el método de Otsu es utilizado para superar esa limitación. En este último método, el umbral óptimo es identificado usando el histograma de la imagen.

3.3.2. Región de crecimiento

Es un procedimiento que agrupa los píxeles de la imagen entera en subregiones basado en un criterio predefinido. En general, se selecciona un grupo de píxeles semilla en la imagen original, se selecciona un conjunto de criterios de similitud y una regla de paro. Las regiones crecen agregando a cada semilla aquellos píxeles vecinos que tienen propiedades similares a la semilla. Finalmente, el crecimiento de regiones se detiene cuando no hay más píxeles que cumplan con el criterio para su inclusión en esa región.

Este método puede segmentar correctamente las regiones que tienen las mismas propiedades aun cuando estas se encuentren espacialmente separadas. Sin embargo, algunos puntos de inicio diferentes pueden no crecer en regiones similares y además, si el criterio de homogeneidad no es seleccionado propiamente las regiones se pueden propagar o unir con regiones adyacentes.

Este método es menos sensible al ruido de la imagen que la umbralización debido al uso de propiedades de las regiones. Sin embargo, pertenece a la clase de algoritmos iterativos, por tanto su desventaja es que necesita mucho tiempo de cómputo.

3.3.3. División y unión de regiones

Al igual que el método anterior, este tipo de técnica de segmentación es menos sensible al ruido que la umbralización, pero también necesita mucho tiempo de

cómputo.

En lugar que el usuario tenga que elegir puntos semilla como en el método anterior, puede dividir una imagen en un conjunto de regiones no conectadas arbitrarias. El segundo paso será entonces unir las regiones en un intento por satisfacer las condiciones que darán lugar a una razonable segmentación. Este método es usualmente implementado con teoría basada en árboles cuadrangulares de datos. Supongamos que R representa la imagen entera y seleccionamos un predicado Q .

1. Iniciando con la imagen entera, si $Q(R) = falso$, dividimos la imagen en cuadrantes. Si Q es falso en algún cuadrante, es decir, si $Q(R_i) = falso$, dividimos ese cuadrante en sub cuadrantes y así sucesivamente hasta que no sea posible ninguna otra división.
2. Si solo se utiliza división, la partición final puede contener regiones adyacentes con propiedades idénticas. Este inconveniente se puede remediar, al permitir la fusión, así como la división, es decir, unir cualesquiera dos regiones adyacentes R_i y R_j para las cuales $Q(R_i \cup R_j) = true$.

3.4. Segmentación basada en teoría especial

En este tipo de técnicas, la segmentación se realiza de manera totalmente diferente a las técnicas anteriormente mencionadas, según el método empleado. Numerosos algoritmos de segmentación basados en teoría especial derivan de otros campos de conocimiento tales como transformaciones de onda, morfología, matemáticas difusas, algoritmos genéticos, inteligencia artificial, entre otras. Y en general, entre otros métodos, se encuentran la segmentación basada en redes neuronales y la segmentación basada en agrupación.

3.4.1. Segmentación basada en redes neuronales

La segmentación basada en redes neuronales es totalmente diferente de los algoritmos convencionales. Primeramente, la imagen debe ser mapeada totalmente a una red neuronal, donde cada neurona representará un pixel. Por tanto, un prob-

lema de segmentación se convierte en un problema de minimización de energía.

La red neuronal es entrenada con un conjunto de datos de entrenamiento con el fin de determinar las conexiones y los pesos entre los nodos. Finalmente, las nuevas imágenes serán segmentadas con estas redes neuronales entrenadas.

Entre sus ventajas se puede observar que este tipo de segmentación tiene alta habilidad paralela y una capacidad de cómputo rápido, lo que lo hace apropiado para aplicaciones en tiempo real. No existen restricciones a grados no lineales y tiene alta interacción entre las unidades de procesamiento, lo que hace a esta algoritmo capaz de establecer modelos para cualquier proceso. La robustez lo hace insensible al ruido. Finalmente, tiene un reducido requerimiento de la intervención de expertos durante el proceso de segmentación de la imagen.

Por otro lado, entre sus desventajas se encuentran que algunos tipos de información de segmentación deben de conocerse de antemano. La inicialización puede influenciar el resultado de la segmentación. La red neuronal debe ser previamente entrenada y el proceso de entrenamiento puede ser muy largo y al mismo tiempo, se debe evitar el sobre entrenamiento.

3.4.2. Segmentación basada en agrupación

La agrupación es una tarea de aprendizaje no supervisado, donde se necesita identificar un conjunto finito de categorías para clasificar píxeles. Este método no utiliza ninguna etapa de entrenamiento, en su lugar entrena con los datos disponibles y se utiliza principalmente cuando las clases se conocen de antemano.

Se define un criterio de similaridad entre los píxeles y entonces los píxeles similares son reunidos para formar las agrupaciones. Este criterio es determinado por medida de distancias tales como la distancia Euclidiana o la distancia Mahalanobis. La reunión de píxeles en grupos se basa en el principio de maximizar la similitud intra clase y maximizar la similitud inter clase. Finalmente, la calidad del resultado depende tanto de la medida de similaridad usada como de su implementación.

3.5. Segmentación basada en modelos

Debido a la presencia de objetos desordenados, texturas, ruido, variaciones en la iluminación, entre otros, los métodos basados en modelos han sido utilizados con éxito considerable para superar todas esas dificultades en la segmentación de imágenes, debido a su capacidad de integrar los conocimientos de alto nivel con la información de procesamiento de imágenes de bajo nivel.

Además, han dominado los campos de visión computacional y procesamiento de imágenes debido a su habilidad de transformar un problema de modelado de segmentación en un sistema de ecuaciones diferencial parciales (PDE's por sus siglas en inglés); su habilidad de incorporar e integrar regularizadores en esos modelos; su capacidad de resolver PDE's utilizando métodos de diferencias finitas (FDM por sus siglas en inglés); su capacidad de vinculación entre PDE's y el sistema de conjuntos de nivel para implementar métodos de diferencias finitas; su capacidad de extender los sistemas de PDE's de 2-D a 3-D o incluso dimensiones superiores; su habilidad para proporcionar soluciones en una forma rápida, estable y cerrada; y por último, su capacidad de manejar interactivamente segmentación de imágenes en el sistema de PDE's.

La idea central de este modelo es transformar un problema de modelado de segmentación en un sistema de ecuaciones diferenciales parciales. Esto es, la evolución de una curva, superficie o imagen es manejada por una ecuación diferencial parcial y la solución de esa ecuación es la segmentación que esperamos.

Los métodos basados en modelos más comunes son los modelos deformables que son clasificados en modelos deformables paramétricos y no paramétricos:

- Los modelos deformables paramétricos o modelos de contornos activos es un modelo de minimización de energía y su implementación resulta en curvas que se mueven dentro de la imagen para encontrar fronteras de objetos bajo la influencia de fuerzas internas o externas. Estos modelos son más rápidos que los modelos deformables no paramétricos y sus limitaciones son pequeños rangos de captura y complejidad de movimiento en fronteras de regiones cóncavas.
- Los modelos deformables no paramétricos o modelos deformables geométri-

cos, fueron propuestos basados en método de conjuntos de nivel y teoría de convolución de curva. Este método realiza la segmentación empezando con una curva inicial y evoluciona su forma usando la ecuación de velocidad.

Los principales métodos basados en PDE son: snakes o contornos activos, conjuntos de nivel, el modelo Mumford-Shah y el modelo Chan-Vese.

3.5.1. Modelo de contornos activos o snakes

Los contornos activos o snakes son curvas generadas por computadora, definidas dentro del dominio de la imagen, que se mueven dentro de la ella para encontrar los límites de los objetos bajo la influencia de fuerzas externas o internas.

Tradicionalmente, un contorno activo es una curva

$$c(s) = [x(s), y(s)], \quad s \in [0, 1]$$

que se mueve a través del dominio espacial de la imagen para minimizar la energía representada por la funcional

$$\begin{aligned} E(c(s)) &= E_{\text{int}}(c(s)) + E_{\text{ext}}(c(s)) \\ &= \int_0^1 \frac{\alpha |c'(s)|^2 + \beta |c''(s)|^2}{2} ds + \int_0^1 g(u(c(s))) ds \end{aligned}$$

donde $E_{\text{int}}(c(s))$, es la energía de deformación interna que caracteriza el contorno, α y β son parámetros de peso que controlan la tensión y rigidez de la curva, respectivamente, y $c'(s)$ y $c''(s)$ denotan la primera y segunda derivada de la curva respecto al parámetro s .

La función de energía externa $E_{\text{ext}}(c(s))$, es derivada de la imagen de tal modo que tome en sus valores más pequeños las características de interés, tales como las fronteras. En este caso, $g(u(c(s)))$, denota una función escalar potencial definida en el plano de la imagen. Algunos ejemplos de estas funciones son:

1. $\pm G_\sigma(x, y) * u(x, y)$ que es utilizado para líneas.

2. $g(u(x, y)) = -|\nabla G_\sigma(x, y) * u(x, y)|^2$ utilizado para bordes escalonados, donde $u(x, y)$ es una imagen en niveles de gris.

donde $G_\sigma(x, y)$ es una función gaussiana en dos dimensiones con desviación estándar σ , y ∇ es el operador gradiente.

Entre las limitaciones de este tipo de modelos se puede encontrar que requieren de la interacción del usuario, que consiste en la determinación de la curva al rededor del objeto que se desea detectar, ya que por un lado son sensibles a las condiciones iniciales y por otro, la función que representa la energía usualmente converge a la energía mínima local. Además, debido a la parametrización explícita del modelo, no se puede hacer frente a partes salientes y a los cambios topológicos. Es sensible al ruido y finalmente, la complejidad computacional es alta.

3.5.2. Modelo de conjuntos de nivel

A pesar de las mejoras realizadas al modelo básico de contornos activos, fueron Osher y Sethian quienes desarrollaron un algoritmo que supero las dificultades presentadas por ese método. Su modelo de conjuntos de nivel fue muy influyente y útil. La idea básica es representar las curvas o superficies como el conjunto de nivel cero de una hiper-superficie de dimensión alta.

Básicamente, esto significa que las curvas cerradas en una superficie 2 - dimensional son consideradas como una superficie continua de un espacio 3 - dimensional. La definición de una función de suavizado $\phi(x, y, t)$ representa la superficie mientras que el conjunto de definiciones $\phi(x, y, t) = 0$ representa las curvas. Entonces la evolución de una curva puede ser transformado en la evolución de una función de conjuntos de nivel en el espacio 3-D.

Sea $\phi(x, y, t = 0)$ una función de conjuntos de nivel, cuyo conjunto de nivel cero corresponde a la curva. Con la curva como frontera, la superficie completa puede dividirse en región interna y región externa referente a esa curva. Se define una función de distancia signada (SDF por sus siglas en inglés) en la superficie

$$\phi(x, y, t = 0) = d$$

donde el valor de d es la distancia más corta entre el punto x sobre la superficie

y la curva. Durante todo el proceso evolutivo de la curva, sus puntos ajustan la fórmula

$$\phi(x, y, t) = 0$$

la fórmula de movimiento común de conjuntos de nivel es

$$\phi_t + F|\nabla\phi| = 0$$

donde F es la velocidad, la cual está relacionada con la evolución de las características de la superficie como curvatura, dirección normal, entre otras, y características de la imagen como el nivel de gris, gradiente, entre otras. Cuando se aplica a segmentación de imágenes, el diseño de F depende de la información de la imagen y el valor ideal es cero en el borde del objeto, es decir, el valor más grande del gradiente de grises.

Este método cuenta con ventajas como su estabilidad e irrelevancia con la topología, muestra una gran ventaja para resolver problemas de producción de puntos esquina, rotura y combinación de curva, entre otros. Además provee más implementaciones numéricas precisas. Sin embargo, es necesario resolver algunos problemas tales como la elección de la función de conjuntos de nivel, la expansión de la función de velocidad, la función de paro en los bordes, entre otros.

Por otro lado, entre las desventajas encontramos que dado que la función de paro de borde depende del gradiente de la imagen, solo objetos con bordes definidos por gradientes pueden ser segmentados. Además, en la práctica, la función de paro de borde nunca es exactamente cero en los bordes, así que la curva eventualmente puede pasar a través del contorno del objeto.

3.5.3. Modelo Mumford-Shah

El modelo Mumford-Shah es un modelo sobresaliente para la segmentación de imágenes, que fue propuesto basado en un sistema variacional en su artículo pionero en 1989. Este modelo utiliza la información global de la imagen como criterio de paro para segmentar la imagen.

Define la curva cerrada (C) en el plano de la imagen (Ω), la cual separa este

plano en dos partes, consideradas como el interior y el exterior de la curva. También define u como el conjunto de datos de la imagen inicial (en 2-D o 3-D) definida en Ω y u_0 como una aproximación suave por partes de u .

La segmentación de la imagen en objetos homogéneos es realizada a través de la búsqueda de un par de objetos (u_0, C) , y la definición óptima para este par de objetos es realizada mediante la minimización de una energía que representa la funcional $F(u_0, C)$. Así, el objetivo de este método es minimizar una función de energía definida como

$$F(u_0, C) = \mu \cdot \text{length}(C) + \iint_{\Omega} |u_0 - u|^2 dx dy + \nu \iint_{\Omega/C} |\nabla u_0|^2 dx dy$$

donde $\mu > 0$, $\nu > 0$ son parámetros fijos. El primer término en esta funcional asegura que las discontinuidades de C tengan longitud mínima, el segundo término asegura que u_0 sea una buena aproximación a u y el tercer término asegura que u_0 sea suave.

Es claro que únicamente cuando el contorno se encuentre en el borde de dos regiones homogéneas F podrá tener el valor mínimo. Finalmente la energía de ajuste es minimizada si $C = C_0$, donde C_0 es el contorno buscado. En consecuencia, este modelo toma ventaja de toda la información de la imagen para resultar en una mejor segmentación de la imagen.

3.5.4. Modelo Chan-Vese

Basados en el trabajo realizado por Mumford y Shah, Chan y Vese desarrollaron una nueva funcional para la segmentación de imágenes basada en la homogeneidad.

La idea básica es la búsqueda de una partición de la imagen dada en dos regiones, una región representa el objeto a ser detectado y la otra el fondo. Este modelo no está basado en la función de borde para detener la curva de evolución en el borde deseado. Puede detectar objetos cuyo borde no necesariamente están definidos por el gradiente o, en otras palabras, no son muy suaves. Iniciando con una sola curva inicial este modelo puede automáticamente detectar contornos

interiores y no necesariamente iniciar alrededor de los objetos que se desean detectar. Más aún, en la implementación de este modelo no se necesita suavizar la imagen inicial, incluso si es muy ruidosa, ya que aún así las ubicaciones de los bordes son muy bien detectados.

Como una idea más formal, se considera un caso simple donde la imagen $I(x, y)$ está formada por dos regiones de intensidades constantes a trozos. Se denotan los valores de las intensidades de esas dos regiones como c_1 y c_2 , y se asume que el objeto que se desea detectar tiene una región cuya frontera es C_0 y su intensidad es c_1 . Entonces dentro de la curva ($inside(C)$), la intensidad de $I(x, y)$ es aproximadamente c_1 , mientras que fuera de ella ($outside(C)$), la intensidad de $I(x, y)$ es aproximadamente c_2 . Así que se considera el término de ajuste

$$\begin{aligned} F(c_1, c_2, C) = & \mu \cdot \text{length}(C) + \nu \cdot \text{area}(inside(C)) \\ & + \lambda_1 \int_{inside(C)} |I(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{outside(C)} |I(x, y) - c_2|^2 dx dy \end{aligned}$$

donde C es una curva, y las constantes c_1 y c_2 son los promedios de $I(x, y)$ dentro y fuera de C respectivamente. Por otro lado, $\text{length}(C)$ es la longitud del borde cerrado, $\text{area}(inside(C))$ es el área del interior de C y $\mu > 0, \nu > 0, \lambda_1$ y λ_2 son parámetros fijos.

Así como en el método de Mumford-Shah, en este método F alcanza el valor mínimo sólo cuando el contorno se encuentra en el borde de las dos regiones homogéneas y de la misma manera, la energía de ajuste es minimizada cuando $C = C_0$, donde C_0 es el borde que se desea encontrar. En consecuencia el modelo simplificado de Mumford-Shah también toma ventaja de toda la información de la imagen para resultar en una mejor segmentación de la imagen.

Capítulo 4

Método de Tsai et al.

4.1. Ecuación de alineación

Los autores Tsai et al.^[31] proponen primeramente un modelo para la alineación de un conjunto de imágenes binarias, lo cual de alguna forma facilitará los procesos posteriores para la segmentación basada en la forma.

Sea \mathcal{T} , el conjunto de n imágenes binarias que servirán para la etapa de entrenamiento:

$$\mathcal{T} = \{I^1, I^2, \dots, I^n\}$$

Dado que el primer objetivo es alinear este conjunto de entrenamiento y debido a que cada imagen puede tener una posición diferente al resto de ellas, es decir, debido a que cada elemento $I^i \in \mathcal{T}$, puede tener una posición respecto al eje x , respecto al eje y , una escala y una rotación diferente a las otras imágenes de entrenamiento, será de vital importancia calcular \mathcal{P} , el conjunto de n parámetros de pose¹:

$$\mathcal{P} = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^n\}$$

Como se podrá observar, cada elemento de \mathcal{P} pertenece a sólo un elemento de \mathcal{T} . Ahora, ya que el conjunto \mathcal{P} se utilizará para eliminar variaciones en la forma, causadas por diferentes poses, cada elemento de ese conjunto, a su vez es

¹Se considera como *pose*, la posición y orientación que permiten describir completamente un cuerpo rígido en el espacio^[31].

un conjunto de 4 parámetros:

$$\mathbf{p}^i = [a^i, b^i, h^i, \theta^i]^T$$

donde a^i, b^i, h^i y θ^i corresponden respectivamente a la traslación en x , traslación en y , escalamiento y rotación de la imagen I^i .

La imagen transformada a partir de I , basada en los parámetros de \mathbf{p} , se denota como \tilde{I} , y es definida como

$$\tilde{I}(\tilde{x}, \tilde{y}) = I(x, y)$$

donde

$$\begin{aligned} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} &= T[\mathbf{p}] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}}_{M(a,b)} \underbrace{\begin{bmatrix} h & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{H(h)} \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(\theta)} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{aligned}$$

Como podrá notarse, la transformación T es un producto de tres matrices: la matriz de traslación $M(a, b)$, la matriz de escalamiento $H(h)$ y la matriz de rotación $R(\theta)$. Por tanto, esta transformación mapea las coordenadas $(x, y) \in \mathbb{R}^2$ a las coordenadas $(\tilde{x}, \tilde{y}) \in \mathbb{R}^2$.

Una forma de alinear las n imágenes binarias, es minimizando la energía representada por la funcional:

$$E_{\text{align}} = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \left\{ \frac{\iint_{\Omega} (\tilde{I}^i - \tilde{I}^j)^2 dA}{\iint_{\Omega} (\tilde{I}^i + \tilde{I}^j)^2 dA} \right\} \quad (4.1)$$

donde Ω denota el dominio de la imagen.

El gradiente de E_{align} , tomado con respecto a \mathbf{p}^i para cualquier i , está dado por

$$\nabla_{\mathbf{p}^i} E_{\text{align}} = \sum_{\substack{j=1 \\ j \neq i}}^n \left\{ \frac{2 \iint_{\Omega} (\tilde{I}^i - \tilde{I}^j) \nabla_{\mathbf{p}^i} \tilde{I}^i dA}{\iint_{\Omega} (\tilde{I}^i + \tilde{I}^j)^2 dA} - \frac{2 \iint_{\Omega} (\tilde{I}^i - \tilde{I}^j)^2 dA \iint_{\Omega} (\tilde{I}^i + \tilde{I}^j) \nabla_{\mathbf{p}^i} \tilde{I}^i dA}{\left(\iint_{\Omega} (\tilde{I}^i + \tilde{I}^j)^2 dA \right)^2} \right\}$$

donde $\nabla_{\mathbf{p}^i} \tilde{I}^i$ es el gradiente de la imagen transformada \tilde{I}^i tomado con respecto al parámetro \mathbf{p}^i . Utilizando la regla de la cadena, el l -ésimo componente de $\nabla_{\mathbf{p}^i} \tilde{I}^i$ está dado por

$$\nabla_{\mathbf{p}^i} \tilde{I}^i(\tilde{x}, \tilde{y}) = \begin{bmatrix} \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{x}} & \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{y}} & 0 \end{bmatrix} \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_l^i} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

donde

$$\begin{aligned} \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_1^i} &= \frac{\partial T[\mathbf{p}^i]}{\partial a^i} = \frac{\partial M(a^i, b^i)}{\partial a^i} H(h^i) R(\theta^i) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_2^i} &= \frac{\partial T[\mathbf{p}^i]}{\partial b^i} = \frac{\partial M(a^i, b^i)}{\partial b^i} H(h^i) R(\theta^i) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_3^i} &= \frac{\partial T[\mathbf{p}^i]}{\partial h^i} = M(a^i, b^i) \frac{\partial H(h^i)}{\partial h^i} R(\theta^i) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_4^i} &= \frac{\partial T[\mathbf{p}^i]}{\partial \theta^i} = M(a^i, b^i) H(h^i) \frac{\partial R(\theta^i)}{\partial \theta^i} = \begin{bmatrix} -h \sin(\theta) & -h \cos(\theta) & 0 \\ h \cos(\theta) & -h \sin(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

por lo tanto, tenemos que:

$$\begin{aligned}\nabla_{\mathbf{p}_1} \tilde{I}^i(\tilde{x}, \tilde{y}) &= \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{x}} \\ \nabla_{\mathbf{p}_2} \tilde{I}^i(\tilde{x}, \tilde{y}) &= \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{y}} \\ \nabla_{\mathbf{p}_3} \tilde{I}^i(\tilde{x}, \tilde{y}) &= \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{x}} (x \cos(\theta) - y \sin(\theta)) + \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{y}} (x \sin(\theta) + y \cos(\theta)) \\ \nabla_{\mathbf{p}_4} \tilde{I}^i(\tilde{x}, \tilde{y}) &= -\frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{x}} (xh \sin(\theta) + yh \cos(\theta)) \\ &\quad + \frac{\partial \tilde{I}^i(\tilde{x}, \tilde{y})}{\partial \tilde{y}} (xh \cos(\theta) - yh \sin(\theta))\end{aligned}$$

4.2. Análisis de la implementación computacional

4.2.1. Error en las transformaciones

Uno de los problemas que se tuvo al realizar el algoritmo de alineamiento, fue las imágenes resultantes después de aplicar las transformaciones a las imágenes de entrenamiento. Este problema se visualiza cuando se realizan operaciones de escalamiento y de rotación.

A fin de ilustrar lo anterior, partamos de la Figura 4.1(a) y observemos el resultado de distintas transformaciones, es decir el resultado de aplicarle diferentes valores de \mathbf{p} :

- Utilizando $\mathbf{p} = [5, 10, 1, 0]$, se obtiene como resultado la ilustración de la Figura 4.1(b), la cual no presenta ninguna alteración.
- Utilizando $\mathbf{p} = [0, 0, 1.2, 0]$, es decir, al realizar un escalamiento del 20%, se produce la alteración de la Figura 4.1(c).
- Si se utiliza $\mathbf{p} = [0, 0, 1, 35]$, es decir, se aplica una rotación de 35°, la alteración obtenida se visualiza en la Figura 4.1(d).
- Al utilizar $\mathbf{p} = [0, 0, 1.2, 35]$, es decir, rotar y escalar al mismo tiempo, se puede notar en la Figura 4.1(e) el resultado de las alteraciones.

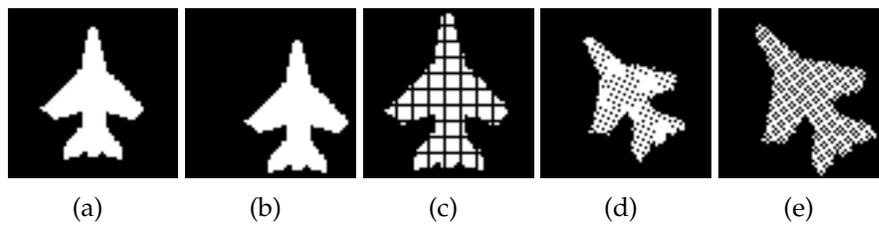


Figura 4.1: Transformación de una imagen: (a) imagen original, (b) traslación en x y y , (c) escalamiento, (d) rotación, (e) escalamiento y rotación.

Una forma rápida y fácil, aunque no con buenos resultados, de resolver los problemas anteriores, es aplicar el filtro de promedio con un kernel de 7×7 , para tratar de rellenar los espacios que no cuentan con información, es decir, las líneas o puntos negros sobre las regiones blancas. El resultado final se muestra en la Figura 4.2.

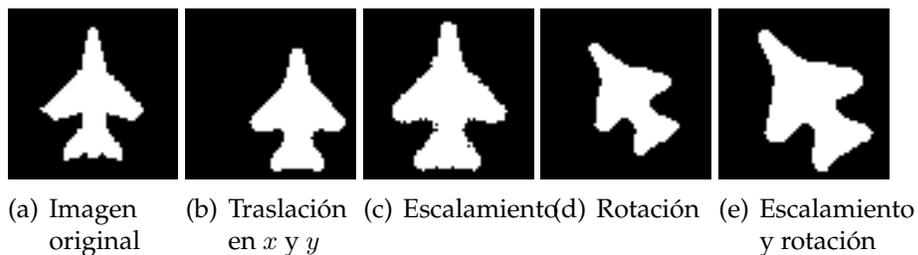


Figura 4.2: Transformación de una imagen y aplicación de un filtro promedio.

Como puede observarse, el resultado es poco favorable ya que se pierde la información del contorno original, el cual es de gran importancia. El caso que más resalta la pérdida de información, se puede apreciar en la Figura 4.2(b), pues aunque no sea necesario, ya que sólo se traslada la imagen, el filtro afecta la imagen resultante. Esto representa un gran problema para la implementación del algoritmo, pues aunque no se realice ningún cambio con respecto a los parámetros de \mathbf{p} (suponiendo $\mathbf{p} = [0, 0, 1, 0]$), este filtro reducirá a la imagen en tamaño y la deformará en cada iteración.

4.2.2. Importancia del orden en la minimización

Una parte importante de la implementación realizada, es que al minimizar la energía 4.1, se debe tener especial cuidado en el orden en el que esta tarea se realiza.

Se observó que cuando se intentaba minimizar la energía utilizando los cuatro parámetros de \mathbf{p} , esto es, utilizando los parámetros a , b , h y θ al mismo tiempo, el algoritmo se ciclaba y por tanto nunca se llegaba a la solución.

Para corregir este problema, el algoritmo se implementó de tal forma que se especificaba primeramente la minimización de los parámetros a y b . Después de obtener los valores óptimos para estos dos parámetros, se proseguía a la minimización del siguiente parámetro h . Durante la ejecución del algoritmo, cuando se modificaba el valor de h , se regresaba al paso anterior, es decir, se volvía a minimizar únicamente los primeros dos parámetros y este proceso se repetía una y otra vez hasta que los tres primeros parámetros tuvieran sus valores óptimos respectivos.

Análogamente se realizó la implementación de la minimización del parámetro θ , después que los tres primeros parámetros ya han sido minimizados, se procede a minimizar este último parámetro y en el transcurso se modifican los parámetros anteriores por lo cual se deben volver a minimizar en el orden descrito.

4.3. Ecuación de la curva implícita

4.3.1. Mapas de distancia

Para poder describir los mapas de distancia, primeramente se debe definir la función implícita, después la función distancia y finalmente la función de distancia signada.

Supongamos que en una dimensión, dividimos la recta real en cinco distintas subregiones utilizando los puntos

$$x = -2, x = -1, x = 1, x = 2$$

definiéndose así los subintervalos

$$(-\infty, -2), (-2, -1), (-1, 1), (1, 2), (2, \infty)$$

A la región o dominio original, es decir la recta real, la llamaremos Ω , y nos referiremos entonces a:

$$\Omega^- = (-2, -1) \cup (1, 2)$$

$$\Omega^+ = (-\infty, -2) \cup (-1, 1) \cup (2, \infty)$$

$$\partial\Omega = \{-2, -1, 1, 2\}$$

respectivamente como el interior del dominio, el exterior del dominio y la interfaz como se ilustra en la figura 4.3.

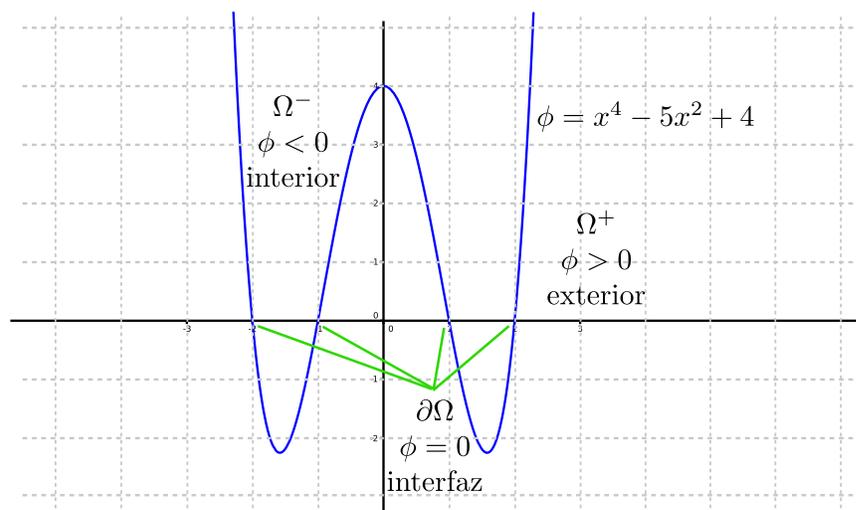


Figura 4.3: Función implícita $\phi = x^4 - 5x^2 + 4$

Otro ejemplo más sencillo como el de la figura 4.4, se puede encontrar en el libro de Osher y Fedkiw^[22]. En este ejemplo, la recta real es dividida en tres distintas subregiones utilizando los puntos $x = -1$ y $x = 1$, definiéndose así los subintervalos $(-\infty, -1)$, $(-1, 1)$ y $(1, \infty)$. Por lo tanto el interior, el exterior y la

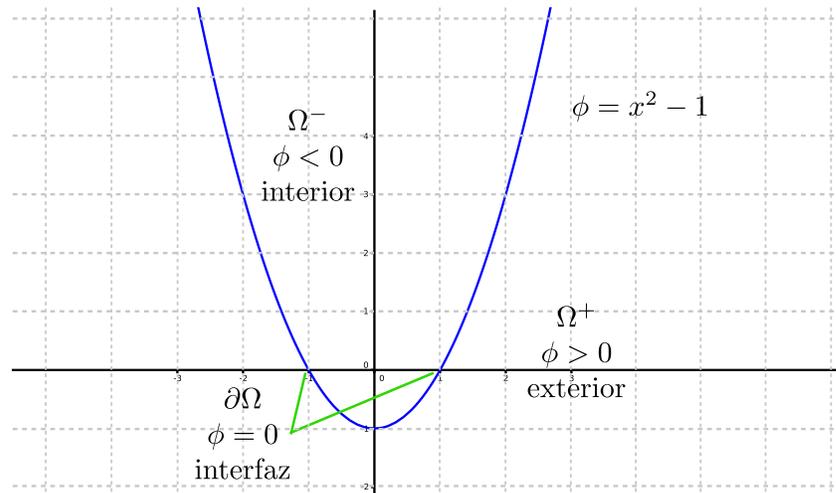


Figura 4.4: Función implícita $\phi = x^2 - 1$

interfaz del dominio serán:

$$\Omega^- = (-1, 1)$$

$$\Omega^+ = (-\infty, -1) \cup (1, \infty)$$

$$\partial\Omega = \{-1, 1\}$$

Analizando estos dos ejemplos se puede concluir que mientras los subdominios tienen la misma dimensión que el dominio, la interfaz tiene una dimensión menor, y en general, en \mathbb{R}^n , mientras los subdominios son n -dimensionales, las interfaz son de dimensión $n - 1$.

Como ya se habrá observado al definir $\partial\Omega$, en una representación de interfaz explícita, se debe expresar el conjunto de puntos completo como en los dos ejemplos anteriores. Sin embargo, para evitar describir expresamente todos los puntos, alternativamente se puede utilizar una *representación de interfaz implícita* que define al conjunto $\partial\Omega$ como el isocontorno de alguna función. Por ejemplo, el isocontorno cero de la función

$$\phi(x) = x^2 - 1$$

es el conjunto de todos los puntos donde $\phi(x) = 0$, es decir, es exactamente el mismo conjunto $\partial\Omega = \{-1, 1\}$. En esta representación, igualmente se puede notar

que la función implícita $\phi(x)$ es definida a través del dominio unidimensional, mientras que el isocontorno que define la interfaz es una dimensión menor. Más generalmente, en \mathbb{R}^n , la función implícita $\phi(\mathbf{x})$ es definida sobre todos los vectores $\mathbf{x} \in \mathbb{R}^n$, y su isocontorno tiene dimensión $n - 1$. Más aún, para cualquier función $\hat{\phi}(\mathbf{x})$ y un arbitrario isocontorno $\hat{\phi}(\mathbf{x}) = a$, para algún escalar $a \in \mathbb{R}$, podemos definir

$$\phi(\mathbf{x}) = \hat{\phi}(\mathbf{x}) - a$$

de tal forma que el isocontorno $\phi(\mathbf{x}) = 0$ de ϕ es idénticamente el isocontorno $\hat{\phi}(\mathbf{x}) = a$ de $\hat{\phi}$.

Por otro lado, en dos dimensiones, la interfaz de dimensión menor, es una curva cerrada que separa el espacio \mathbb{R}^2 en subdominios separados con área distinta de cero. Considerando por ejemplo la función

$$\phi(\mathbf{x}) = x^2 + y^2 - 1$$

el isocontorno $\phi(\mathbf{x}) = 0$, define la interfaz como el círculo unitario

$$\partial\Omega = \{\mathbf{x} : |\mathbf{x}| = 1\}$$

la región interior como el disco abierto

$$\Omega^- = \{\mathbf{x} : |\mathbf{x}| < 1\}$$

y la región exterior como

$$\Omega^+ = \{\mathbf{x} : |\mathbf{x}| > 1\}$$

Para finalizar con este bloque referente a la función ϕ , es necesario describir algunas herramientas geométricas muy útiles, consecuencia de esta forma de representar implícitamente la interfaz^[22].

El gradiente de la función implícita en \mathbb{R}^n , es definido como:

$$\nabla\phi = \left(\frac{\partial\phi}{\partial x_1}, \dots, \frac{\partial\phi}{\partial x_n} \right)$$

el cual es perpendicular al isocontorno de ϕ y puntos en la dirección de incre-

mento de ϕ . De esta forma, si $\nabla\phi$ es evaluado en algún punto perteneciente al isocontorno cero de ϕ , entonces tendrá la misma dirección que la normal unitaria local \mathbf{N} a la interfaz. Así la normal unitaria es:

$$\mathbf{N} = \frac{\nabla\phi}{|\nabla\phi|} \quad (4.2)$$

para puntos en la interfaz.

Si se define la función Heaviside como:

$$H(\phi) = \begin{cases} 0, & \text{si } \phi \leq 0 \\ 1, & \text{si } \phi > 0 \end{cases} \quad (4.3)$$

entonces la integral de volumen en \mathbb{R}^3 , o respectivamente la integral de área o longitud en \mathbb{R}^2 o \mathbb{R} , de una función sobre el interior de la región Ω^- es definida como:

$$\int_{\Omega} f(\mathbf{x})(1 - H(\phi(\mathbf{x})))$$

de manera similar, la integral de f sobre el exterior de la región Ω^+ es definido como:

$$\int_{\Omega} f(\mathbf{x})H(\phi(\mathbf{x}))$$

Por definición, la derivada direccional de la función Heaviside H en la dirección normal \mathbf{N} , es la función delta de Dirac:

$$\widehat{\delta}(\mathbf{x}) = \nabla H(\phi(\mathbf{x})) \cdot \mathbf{N}$$

luego utilizando la regla de la cadena para tomar el gradiente de H , la definición de la normal de la ecuación (4.2), y el hecho de que $\nabla\phi(\mathbf{x}) \cdot \nabla\phi(\mathbf{x}) = |\nabla\phi(\mathbf{x})|^2$, se puede reescribir la ecuación anterior como:

$$\begin{aligned} \widehat{\delta}(\mathbf{x}) &= H'(\phi(\mathbf{x}))\nabla\phi(\mathbf{x}) \cdot \frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|} \\ &= H'(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})| \end{aligned}$$

y debido a que $\delta(\phi) = H'(\phi)$, la ecuación anterior se reescribe como:

$$\widehat{\delta}(\mathbf{x}) = \delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|$$

Ahora la integral de superficie (respectivamente la integral de línea o punto en \mathbb{R}^2 o \mathbb{R}), de una función f sobre el contorno $\partial\Omega$ es definida como:

$$\int_{\Omega} f(\mathbf{x})\widehat{\delta}(\mathbf{x})d\mathbf{x} = \int_{\Omega} f(\mathbf{x})\delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|d\mathbf{x} \quad (4.4)$$

donde la región de integración es todo Ω .

Continuando con el esquema descrito en las primeras líneas de este apartado, una **función distancia** en \mathbb{R}^n , es definida como:

$$d(\mathbf{x}) = \min(|\mathbf{x} - \mathbf{x}_i|), \text{ para todo } \mathbf{x}_i \in \partial\Omega$$

y esta definición implica que para cualquier punto $\mathbf{x} \in \partial\Omega$, la distancia $d(\mathbf{x}) = 0$. Además se cumple que:

$$|\nabla d| = 1$$

Para demostrar esto, primeramente se describe cómo se puede construir geoméricamente la función distancia:

$$d(\mathbf{x}) = \begin{cases} 0, & \text{si } \mathbf{x} \in \partial\Omega \\ |\mathbf{x} - \mathbf{x}^c|, & \text{si } \mathbf{x} \notin \partial\Omega \end{cases}$$

donde \mathbf{x}^c es el punto en el conjunto $\partial\Omega$, más cercano a \mathbf{x} . Luego, dado que se cumplen las ecuaciones:

$$|\mathbf{x} - \mathbf{x}^c| = \sqrt{(x_1 - x_1^c)^2 + \cdots + (x_n - x_n^c)^2}$$

$$\frac{\partial d(\mathbf{x})}{\partial x_i} = \frac{x_i - x_i^c}{\sqrt{(x_1 - x_1^c)^2 + \cdots + (x_n - x_n^c)^2}}$$

$$\nabla d(\mathbf{x}) = \left(\frac{\partial d(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial d(\mathbf{x})}{\partial x_n} \right)$$

entonces se puede verificar que:

$$\begin{aligned} |\nabla d(\mathbf{x})| &= \sqrt{\left(\frac{\partial d(\mathbf{x})}{\partial x_1}\right)^2 + \dots + \left(\frac{\partial d(\mathbf{x})}{\partial x_n}\right)^2} \\ &= \sqrt{\sum_{i=1}^n \frac{(x_i - x_i^c)^2}{(x_1 - x_1^c)^2 + \dots + (x_n - x_n^c)^2}} \\ &= 1 \end{aligned}$$

El último concepto necesario antes de alcanzar el objetivo de este apartado, es la noción de *función de distancia signada*, la cual es una función implícita ϕ con $|\phi(\mathbf{x})| = d(\mathbf{x})$. Esto es:

$$\phi(\mathbf{x}) = \begin{cases} d(\mathbf{x}) = 0, & \text{si } \mathbf{x} \in \partial\Omega \\ -d(\mathbf{x}), & \text{si } \mathbf{x} \in \Omega^- \\ d(\mathbf{x}), & \text{si } \mathbf{x} \in \Omega^+ \end{cases}$$

Adicionalmente, dado que ϕ está definido en función de d , se tiene que:

$$|\nabla\phi| = 1$$

Ahora, un *mapa de distancia* es el resultado de aplicar la función de distancia signada, en el caso 2D, a una imagen, obteniéndose así una matriz cuyos pixeles tienen valores negativos y positivos en las regiones correspondientes al interior y exterior del contorno respectivamente, como queda ilustrado en la figura 4.5.

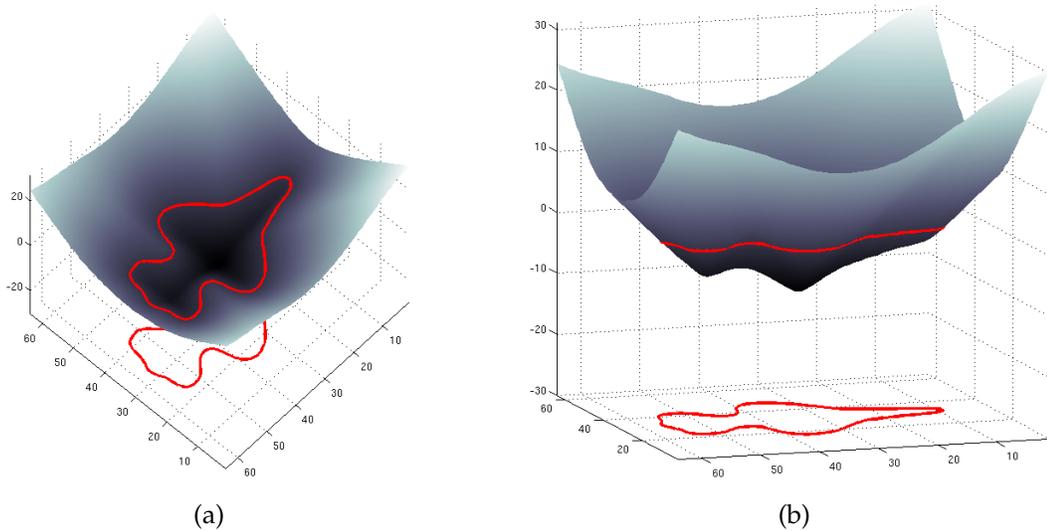


Figura 4.5: Representación gráfica de un mapa de distancias.

4.3.2. Extracción de descriptores

Para extraer las características de una imagen o los descriptores, se utilizan las n imágenes binarias que se mencionan al principio de este capítulo y se realizan los siguientes pasos:

- Se genera el conjunto

$$\{\Psi_1, \Psi_2, \dots, \Psi_n\} \quad (4.5)$$

donde cada Ψ_i representa la función de distancia signada aplicada a la correspondiente imagen I_i . Si I_i es de dimensión $M \times N$, entonces cada Ψ_i también tendrá la misma dimensión.

- Se crea la matriz de dimensión $M \times N$:

$$\bar{\Phi} = \frac{1}{n} \sum \Psi_i$$

la cual representa, como se puede observar, la media de las funciones de distancia signada o funciones de conjuntos de nivel.

- Se obtiene un nuevo conjunto

$$\{\tilde{\Psi}_1, \tilde{\Psi}_2, \dots, \tilde{\Psi}_n\}$$

que es el resultado de sustraer $\bar{\Phi}$ a cada elemento Ψ_i del conjunto (4.5). De nuevo cada elemento de este conjunto tendrá dimensión $M \times N$.

- Dado el conjunto anterior, se genera la matriz

$$\mathcal{S} = \begin{bmatrix} \tilde{\psi}_1 & \tilde{\psi}_2 & \dots & \tilde{\psi}_n \end{bmatrix}$$

donde cada $\tilde{\psi}_i$ es el reordenamiento de $\tilde{\Psi}_i$ en forma de vector. Esto se realiza apilando cada columna de $\tilde{\Psi}_i$ una sobre otra hasta formar un vector columna. Luego \mathcal{S} tiene dimensión $K \times n$, donde $K = M \times N$.

- Se realiza una descomposición en eigenvalores del factor

$$\frac{1}{n} \mathcal{S} \mathcal{S}^T = U \Sigma U^T \quad (4.6)$$

de dimensión $K \times K$, donde U es una matriz de $K \times n$, cuyas columnas representan los n modos ortogonales de variación en la forma y Σ es una matriz diagonal de $n \times n$ cuyos elementos de la diagonal representan los correspondientes eigenvalores no cero.

- Al tomar U_i , la i -ésima columna de U , y reordenar los K elementos de nuevo en una estructura de dimensión $M \times N$, se obtiene Φ_i , el i -ésimo modo principal o eigenforma. Luego se crea el conjunto:

$$\{\Phi_1, \Phi_2, \dots, \Phi_n\}$$

de las n eigenformas.

Es importante hacer notar que, dado que la dimensión del factor en la ecuación (4.6) en muchos casos es grande ($K \times K$), el cálculo de los eigenvectores y eigenvalores de esta matriz es costoso computacionalmente. Es por esto que una manera

eficiente de realizar dichos cálculos, es mediante la matriz de dimensión $n \times n$ dada por:

$$\mathcal{W} = \frac{1}{n} \mathcal{S}^T \mathcal{S}$$

Si \mathbf{d} es un eigenvector de \mathcal{W} con su correspondiente eigenvalor λ , entonces $\mathcal{S}\mathbf{d}$ es un eigenvector de la ecuación (4.6) con eigenvalor λ .

Finalmente, la descripción implícita de forma está dada por el conjunto de nivel cero de la siguiente ecuación:

$$\Phi[\mathbf{w}, \mathbf{p}](x, y) = \bar{\Phi}(\tilde{x}, \tilde{y}) + \sum_{i=1}^k w_i \Phi_i(\tilde{x}, \tilde{y})$$

donde $\mathbf{w} = \{w_1, w_2, \dots, w_k\}$ son los pesos para las k eigenformas con las varianzas de esos pesos $\{\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2\}$ dadas por los eigenvalores calculados anteriormente. La figura 4.6 muestra el resultado de la implementación de la ecuación

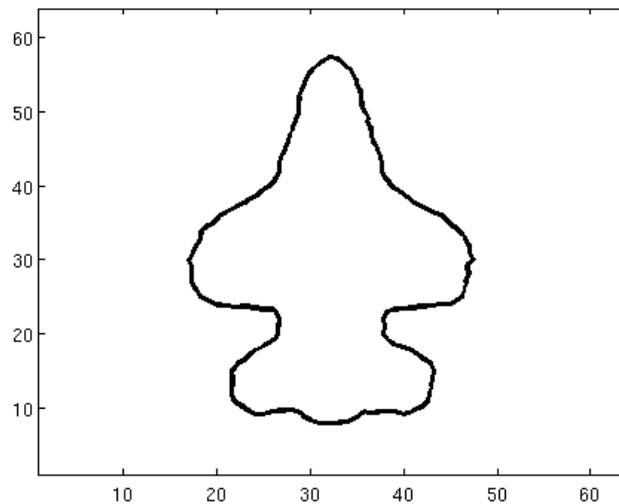


Figura 4.6: Implementación de la ecuación de descripción implícita de forma.

anterior. Esta curva, representa la curva inicial que mediante un proceso iterativo se deformará y dará lugar a la curva que describa el contorno del objeto buscado.

4.4. Adaptación de Chan-Vese

En su trabajo referente a la segmentación de imágenes, Chan y Vese^[4] asumen que la imagen u_0 está formada por dos regiones de intensidades aproximadamente constantes a trozos, de distintos valores u_0^i y u_0^o , y que el objeto que se desea ser detectado, está representado por la región con el valor de intensidades u_0^i . Es por eso que inicialmente consideran la energía:

$$\int_{inside(C)} |u_0 - c_1|^2 dA + \int_{outside(C)} |u_0 - c_2|^2 dA \quad (4.7)$$

donde C es una curva que al ser minimizada, representa el contorno de la región que se desea detectar y las constantes c_1, c_2 , dependientes de C , son los promedios de u_0 dentro de C y respectivamente fuera de C . Sin embargo, en su modelo de contornos activos final, son agregados términos de regularización como la longitud de C y/o el área dentro de C , para obtener:

$$\begin{aligned} F(C, c_1, c_2) &= \mu \cdot (\text{length}(C))^p + \nu \cdot \text{area}(inside(C)) \\ &+ \lambda_1 \int_{inside(C)} |u_0 - c_1|^2 dA \\ &+ \lambda_2 \int_{outside(C)} |u_0 - c_2|^2 dA \end{aligned} \quad (4.8)$$

donde ahora $\mu \geq 0, \nu \geq 0, \lambda_1 > 0, \lambda_2 > 0$ y $p > 0$. Aún así, en sus implementaciones, se utilizan los valores $\lambda_1 = \lambda_2 = 1, \nu = 0$ y $p = 1$.

Utilizando la formulación en conjuntos de nivel, la curva C es representada como el isocontorno cero de la función $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$, de tal forma que:

$$\begin{aligned} C &= \{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}) = 0\} \\ inside(C) &= \{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}) > 0\} \\ outside(C) &= \{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}) < 0\} \end{aligned} \quad (4.9)$$

Usando la función Heaviside H y la medida de Dirac unidimensional δ centrada en 0, definidos respectivamente como:

$$H(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad \text{y} \quad \delta = \frac{d}{dz}H(z)$$

podemos expresar los términos de la ecuación (4.8) de la siguiente forma:

$$\begin{aligned} \text{length}\{\phi = 0\} &= \int_{\Omega} |\nabla H(\phi)| dA = \int_{\Omega} \delta(\phi) |\nabla \phi| dA \\ \text{area}\{\phi \geq 0\} &= \int_{\Omega} H(\phi) dA \\ \int_{\phi \geq 0} |u_0 - c_1|^2 dA &= \int_{\Omega} |u_0 - c_1|^2 H(\phi) dA \\ \int_{\phi < 0} |u_0 - c_2|^2 dA &= \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dA \end{aligned} \tag{4.10}$$

Entonces la energía $F(\phi, c_1, c_2)$, puede ser escrita como:

$$\begin{aligned} F(\phi, c_1, c_2) &= \mu \left(\int_{\Omega} \delta(\phi) |\nabla \phi| dA \right)^p + \nu \int_{\Omega} H(\phi) dA \\ &+ \lambda_1 \int_{\Omega} |u_0 - c_1|^2 H(\phi) dA + \lambda_2 \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dA \end{aligned}$$

donde, fijando ϕ y minimizando la ecuación anterior con respecto a las constantes c_1 y c_2 , éstas se pueden expresar en función de ϕ de la siguiente manera:

$$c_1(\phi) = \frac{\int_{\Omega} u_0 H(\phi) dA}{\int_{\Omega} H(\phi) dA} \quad \text{y} \quad c_2(\phi) = \frac{\int_{\Omega} u_0 (1 - H(\phi)) dA}{\int_{\Omega} (1 - H(\phi)) dA}$$

que representan el promedio de u_0 en $\{\phi \geq 0\}$ y el promedio de u_0 en $\{\phi < 0\}$.

Por su parte, para segmentar una imagen, Tsai et al.^[31] representan a la curva C , como el conjunto de nivel cero de Φ , es decir,

$$C = \{\mathbf{x} \in \mathbb{R}^2 : \Phi(\mathbf{x}) = 0\}$$

Además, como resultado de esa representación implícita, las regiones dentro y fuera de la curva, están dadas respectivamente como:

$$R^u = \{\mathbf{x} \in \mathbb{R}^2 : \Phi(\mathbf{x}) < 0\} \quad \text{y} \quad R^v = \{\mathbf{x} \in \mathbb{R}^2 : \Phi(\mathbf{x}) > 0\}$$

De manera similar a como se definieron las expresiones en la ecuación (4.10), para este caso tenemos:

$$\begin{aligned} A_u &= \int_{\Omega} H(-\Phi) dA & A_v &= \int_{\Omega} H(\Phi) dA \\ S_u &= \int_{\Omega} IH(-\Phi) dA & S_v &= \int_{\Omega} IH(\Phi) dA \\ Q_u &= \int_{\Omega} I^2 H(-\Phi) dA & Q_v &= \int_{\Omega} I^2 H(\Phi) dA \\ \mu &= \frac{S_u}{A_u} & \nu &= \frac{S_v}{A_v} \\ \sigma_u^2 &= \frac{Q_u}{A_u} - \mu^2 & \sigma_v^2 &= \frac{Q_v}{A_v} - \nu^2 \end{aligned}$$

donde A_u, S_u, Q_u, μ y σ_u^2 representan el área, la suma de intensidades, la suma de los cuadrados de las intensidades, el promedio de intensidades y la varianza de la muestra en R^u , donde de igual manera A_v, S_v, Q_v, ν y σ_v^2 representan las mismas estadísticas de la imagen, pero en R^v , y donde la función Heaviside H está dada por la siguiente expresión:

$$H(\Phi) = \begin{cases} 1, & \Phi \geq 0 \\ 0, & \Phi < 0 \end{cases}$$

Así, el modelo propuesto por Chan y Vese^[4] en la ecuación (4.7), se reescribe como:

$$\begin{aligned} E_{CV} &= \int_{R^u} (I - \mu)^2 dA + \int_{R^v} (I - \nu)^2 dA \\ &= \int_{R^u} (I^2 - 2\mu I + \mu^2) H(-\Phi) dA + \int_{R^v} (I^2 - 2\nu I + \nu^2) H(\Phi) dA \\ &= \int_{R^u} I^2 H(-\Phi) dA - 2\mu \int_{R^u} I H(-\Phi) dA + \mu^2 \int_{R^u} H(-\Phi) dA \\ &\quad + \int_{R^v} I^2 H(\Phi) dA - 2\nu \int_{R^v} I H(\Phi) dA + \nu^2 \int_{R^v} H(\Phi) dA \end{aligned}$$

luego, en el caso de las imágenes binarias, mediante la ecuación:

$$\int_{R^u} I^2 H(-\Phi) dA + \int_{R^v} I^2 H(\Phi) dA$$

únicamente se obtiene la suma de las intensidades de la imagen completa, lo cual no representa nada útil al no depender de la curva. Es así que estos términos se desprecian de la ecuación anterior y se obtiene la siguiente expresión:

$$\begin{aligned} E_{CV} &= -2\mu \int_{R^u} I H(-\Phi) dA + \mu^2 \int_{R^u} H(-\Phi) dA \\ &\quad - 2\nu \int_{R^v} I H(\Phi) dA + \nu^2 \int_{R^v} H(\Phi) dA \\ &= (-2\mu S_u + \mu^2 A_u) + (-2\nu S_v + \nu^2 A_v) \end{aligned}$$

Ahora basta notar que:

$$\begin{aligned}
 -2\mu S_u + \mu^2 A_u &= (-2\mu S_u + \mu^2 A_u) \cdot \frac{A_u}{A_u} \\
 &= (-2\mu^2 + \mu^2) \cdot A_u \\
 &= -\mu^2 A_u
 \end{aligned}$$

y análogamente que

$$-2\nu S_v + \nu^2 A_v = -\nu^2 A_v$$

y se tiene por lo tanto:

$$E_{CV} = -(\mu^2 A_u + \nu^2 A_v) = -\left(\frac{S_u^2}{A_u} + \frac{S_v^2}{A_v}\right)$$

Luego se emplea el gradiente descendente para encontrar los parámetros \mathbf{w} y \mathbf{p} que minimicen E_{CV} , para implícitamente determinar la curva de segmentación. Utilizando la regla de la cadena y la regla para derivar cocientes, los gradientes de E_{CV} , tomados con respecto a \mathbf{w} y \mathbf{p} , están dados por las dos ecuaciones a continuación:

$$\begin{aligned}
 \nabla_{\mathbf{w}} E_{CV} &= -\left(\frac{2S_u \cdot \nabla_{\mathbf{w}} S_u \cdot A_u - S_u^2 \cdot \nabla_{\mathbf{w}} A_u}{A_u^2} + \frac{2S_v \cdot \nabla_{\mathbf{w}} S_v \cdot A_v - S_v^2 \cdot \nabla_{\mathbf{w}} A_v}{A_v^2}\right) \\
 &= -\left[(2\mu \nabla_{\mathbf{w}} S_u - \mu^2 \nabla_{\mathbf{w}} A_u) + (2\nu \nabla_{\mathbf{w}} S_v - \nu^2 \nabla_{\mathbf{w}} A_v)\right] \\
 &= -2(\mu \nabla_{\mathbf{w}} S_u + \nu \nabla_{\mathbf{w}} S_v) + (\mu^2 \nabla_{\mathbf{w}} A_u + \nu^2 \nabla_{\mathbf{w}} A_v) \\
 \nabla_{\mathbf{p}} E_{CV} &= -2(\mu \nabla_{\mathbf{p}} S_u + \nu \nabla_{\mathbf{p}} S_v) + (\mu^2 \nabla_{\mathbf{p}} A_u + \nu^2 \nabla_{\mathbf{p}} A_v)
 \end{aligned}$$

Por otro lado, definiendo la medida de Dirac unidimensional δ centrada en cero como:

$$\delta(z) = \frac{d}{dz}H(z)$$

se puede definir el i -ésimo componente de cada uno de los gradientes en las ecuaciones anteriores. Por ejemplo, de la expresión:

$$\nabla_{\mathbf{w}}A_u = \left(\frac{\partial A_u}{\partial w_1}, \dots, \frac{\partial A_u}{\partial w_k} \right)$$

uno de sus componentes puede calcularse como:

$$\begin{aligned} \frac{\partial A_u}{\partial w_i} &= \frac{\partial}{\partial w_i} \iint_{\Omega} H(-\Phi) dA \\ &= \iint_{\Omega} \frac{\partial H(-\Phi)}{\partial w_i} dA \\ &= \iint_{\Omega} \delta(-\Phi) \frac{\partial(-\Phi)}{\partial w_i} dA \\ &= - \iint_{\Omega} \delta(-\Phi) \Phi_i dA \end{aligned}$$

luego de la ecuación (4.4), obtenemos la integral de línea:

$$\frac{\partial A_u}{\partial w_i} = - \oint_C \Phi_i ds$$

Continuando con un razonamiento análogo, se puede verificar que:

$$\begin{aligned} \frac{\partial A_u}{\partial w_i} &= - \frac{\partial A_v}{\partial w_i} = - \oint_C \Phi_i ds \\ \frac{\partial A_u}{\partial p_i} &= - \frac{\partial A_v}{\partial p_i} = - \oint_C \nabla_{p_i} \Phi ds \end{aligned}$$

$$\frac{\partial S_u}{\partial w_i} = -\frac{\partial S_v}{\partial w_i} = -\oint_C I \Phi_i ds$$

$$\frac{\partial S_u}{\partial p_i} = -\frac{\partial S_v}{\partial p_i} = -\oint_C I \nabla_{p_i} \Phi ds$$

donde

$$\nabla_{\mathbf{p}_i} \Phi = \nabla_{\mathbf{p}_i} \Phi(\tilde{x}, \tilde{y}) = \begin{bmatrix} \frac{\partial \Phi(\tilde{x}, \tilde{y})}{\partial \tilde{x}} & \frac{\partial \Phi(\tilde{x}, \tilde{y})}{\partial \tilde{y}} & 0 \end{bmatrix} \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_i} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

con $\frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}_i}$ definido con anterioridad.

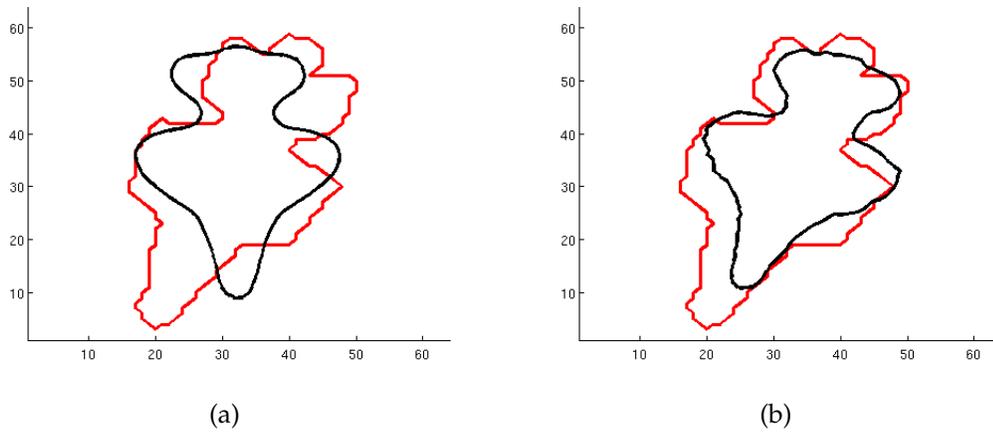


Figura 4.7: Resultado de la implementación: (a) curva inicial, (b) curva después de 400 iteraciones.

La figura 4.7 ilustra el resultado de implementar un algoritmo iterativo, específicamente el descenso del gradiente, que mediante los gradientes de la energía E_{CV} respecto a w y p , permite deformar la curva inicial (curva negra) y obtener la forma del objeto de interés (curva roja).

Capítulo 5

Resultados y aportación

5.1. Introducción

Este apartado se centra en presentar el conjunto de resultados e imágenes obtenidas durante el proceso de desarrollo de las partes más importantes de este trabajo, además de los cambios realizados a la implementación y sus implicaciones.

A modo de resumen, se presentan los resultados obtenidos al calcular la curva inicial que como se mencionó en capítulos anteriores, servirá como punto de partida para luego modificar y encontrar el objeto de interés. Seguidamente se presenta la modificación realizada a esta implementación que resulta en un algoritmo más veloz y finalmente se expone la modificación realizada al modelo de Chan-Vese.

Para el propósito de este apartado, todas las pruebas junto con sus resultados, fueron realizadas en una computadora tipo laptop con las siguientes características:

- Debian

Versión 7.6 (wheezy) de 64-bit

Núcleo Linux 3.2.0-4-amd64

GNOME 3.4.2

- Hardware

Memoria: 3.7 GiB

Procesador: Intel Core i3 CPU M 380 @ 2.53GHz × 4

Además, las imágenes iniciales utilizadas, fueron las siguientes:

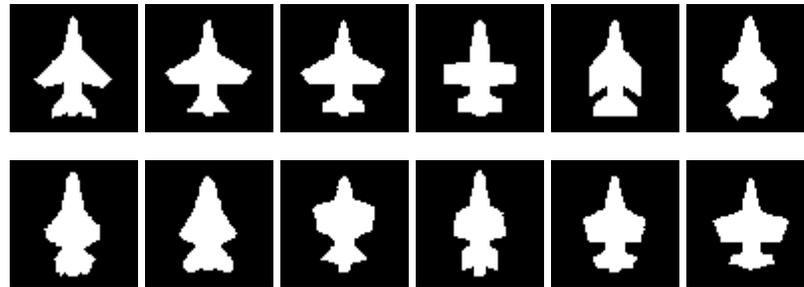


Figura 5.1: Imágenes iniciales.

5.2. Curva inicial

Utilizando el conjunto de imágenes presentado anteriormente y el conjunto de algoritmos que se puede encontrar en la página de MathWorks.com¹, se realizó el cálculo de las funciones de distancia signada. Como resultado, se obtienen las gráficas de la figura 5.2 en la siguiente página.

Estas nuevas matrices, representan los mapas de distancia signada de cada una de las imágenes iniciales. Posteriormente son utilizadas para generar otra matriz como se sugirió en el capítulo anterior, la cual es utilizada para calcular una descomposición SVD.

Como se mencionó con anterioridad, la descripción implícita de forma está dada por el conjunto de nivel cero de la ecuación

$$\Phi[\mathbf{w}, \mathbf{p}](x, y) = \bar{\Phi}(\tilde{x}, \tilde{y}) + \sum_{i=1}^k w_i \Phi_i(\tilde{x}, \tilde{y})$$

donde se puede observar la forma promedio, cuyo contorno se presenta en la figura 4.6, las eigenformas calculadas con la descomposición SVD y los pesos asigna-

¹<http://www.mathworks.com/matlabcentral/fileexchange/submissions/34233/v/2/download/zip>

dos a cada eigenforma. Como se podrá observar sólo se utilizan k eigenformas ($k = 6$ para este caso), y no existe un método para determinar el valor más adecuado para esta variable.

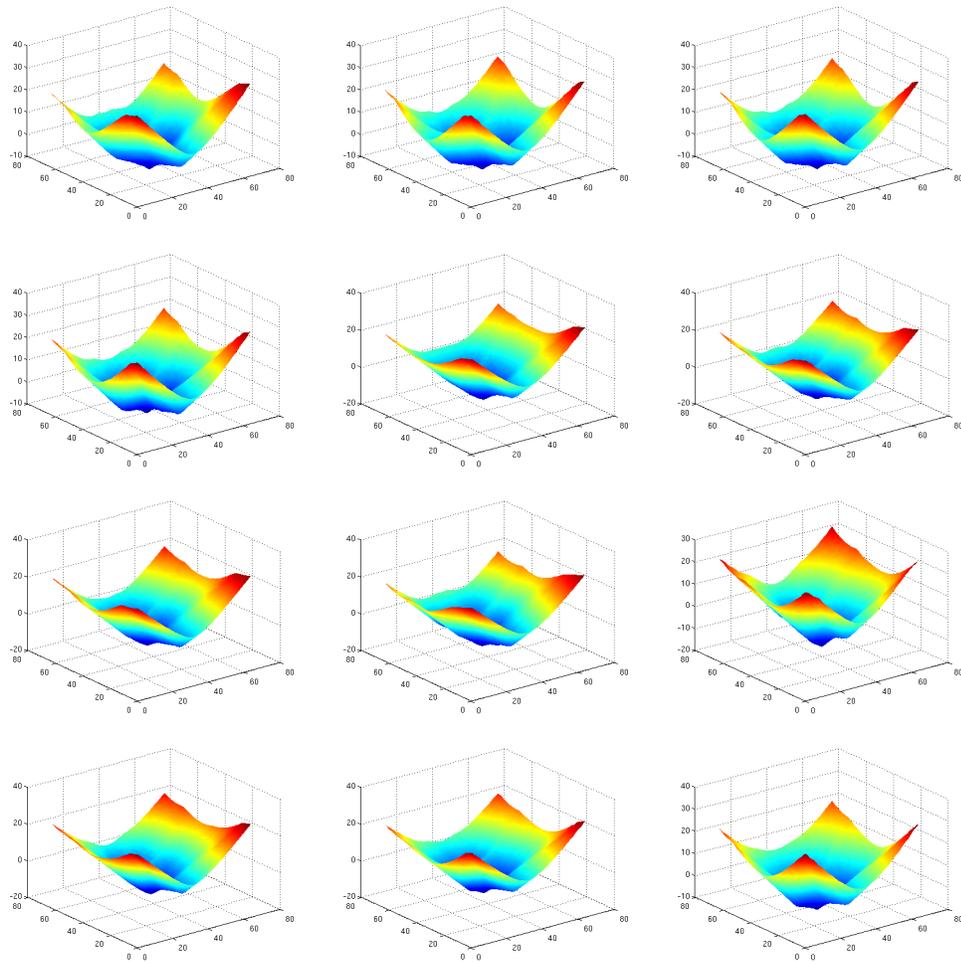


Figura 5.2: Conjunto de mapas de distancia signada para cada imagen de entrada inicial.

Durante del cálculo de las eigenformas, se utilizaron las tres siguientes instrucciones en Matlab

```
profile on;  
profile off;  
profile viewer;
```

como resultado, se obtuvieron los tiempos empleados por cada uno de los comandos del software en la implementación y que se resumen en la siguiente tabla.

Function Name	Calls	Total Time	Self Time
SDF	1	194.94600 s	0.00530 s
reinit_SD	12	194.94069 s	0.15217 s
evolve_normal_WENO	2400	194.67040 s	10.79451 s
der_WENO_minus	307200	85.17593 s	85.17593 s
der_WENO_plus	307200	84.26064 s	84.26064 s
select_der_normal	307200	14.43932 s	14.43932 s
get_dt_normal	2400	0.11568 s	0.11568 s
cargaImágenes	1	0.09593 s	0.01023 s
imread	12	0.07222 s	0.00837 s
addpath	1	0.06092 s	0.00676 s
path	1	0.05050 s	0.03937 s
imagesci/private/readpng	12	0.05049 s	0.02066 s
ismember	12	0.02694 s	0.00406 s
ismember>ismemberlegacy	12	0.02288 s	0.02288 s
strcat	17	0.01348 s	0.01305 s
imagesci/private/imftype	12	0.01211 s	0.00366 s
general/private/parsedirs	2	0.01113 s	0.01078 s
eigenVectors	1	0.00631 s	0.00536 s
imformats	12	0.00602 s	0.00234 s
imformats>find_in_registry	12	0.00368 s	0.00368 s
general/private/catdirs	1	0.00365 s	0.00327 s
imagesci/private/pngreadc	12	0.00245 s	0.00245 s
init_normal_WENO	12	0.00244 s	0.00244 s
imagesci/private/ispng	12	0.00243 s	0.00243 s
imread>parse_inputs	12	0.00125 s	0.00125 s
fliplr	24	0.00116 s	0.00116 s

Tabla 5.1: Nombre, número de ejecuciones, tiempo total y self time (tiempo puro)² de las funciones empleadas en el cálculo de las eigenformas utilizando la función ϕ .

²Tiempo utilizado en una función excluyendo el tiempo empleado en sus funciones hijas.

Es importante comentar que antes de las tres instrucciones anteriores, se utilizaron en consola los comandos

```
ps -C MATLAB
taskset -pc 0 PID
```

y en Matlab los comandos

```
profile -timer real
profile -timer cpu
```

que son necesarios para un correcto funcionamiento del cálculo de tiempos.

Por otro lado, de la tabla se puede observar que el proceso del cálculo de mapas de distancia (filas 3-6), es muy costoso respecto al tiempo, resultando en un tiempo total de más de 3 minutos.

5.3. Función de reemplazo

En su trabajo, Mahmoodi^[19] propone utilizar la función

$$\chi : \Omega \longrightarrow [0, 1]$$

como reemplazo de la función ϕ , es decir la función de distancia signada, en la versión de conjuntos de nivel del modelo propuesto por Mumford-Shah. Los términos de esta versión se pueden apreciar en la ecuación (4.10) y la ecuación resultante después del reemplazo se puede escribir como

$$\begin{aligned} E(c_1, c_2, \chi) = & \mu \iint_{\Omega} |\nabla \chi| dx dy \\ & + \lambda \iint_{\Omega} |u_0(x, y) - c_1|^2 \chi(x, y) dx dy \\ & + \nu \iint_{\Omega} |u_0(x, y) - c_2|^2 (1 - \chi(x, y)) dx dy \end{aligned}$$

donde

$$\chi(x, y) = H(\phi(x, y)) = \frac{1}{\pi} \arctan \frac{\phi(x, y)}{\eta} + \frac{1}{2}$$

y donde η es un parámetro de regularización y H es la función Heaviside.

Según el autor, mediante la implementación de esta función se acelera el proceso de segmentación y al mismo tiempo se disfruta de las mismas propiedades asociadas con el método de conjuntos de nivel, pero sin utilizar la función de distancia signada.

Es así que de esta aproximación, surge la idea de reemplazar todos los usos de la función ϕ en las implementaciones realizadas y el primer cambio, se realiza en el cálculo de las eigenformas que como se mencionó en el apartado anterior, utiliza un conjunto de algoritmos descargados de la página de MathWorks.com para el cálculo de la función de distancia signada.

Sin embargo, en lugar de utilizar la idea propuesta por Mahmoodi^[19], se considera conveniente realizar un híbrido entre la función de distancia signada, la función propuesta y la función Heaviside, resultando finalmente la función

$$\xi : \Omega \longrightarrow \{-1, 0, 1\}$$

Se considera como híbrido ya que conserva la idea de números negativos en la parte interna de la curva, números positivos en la parte externa y el valor cero en la curva (características de la función de distancia signada), sin embargo el rango de números se mantiene entre -1 y 1 (característica similar a la función propuesta χ con imagen en $[0,1]$), y además existen saltos formando escalones entre los segmentos de la imagen (característica de la función Heaviside).

De esta forma los conjuntos expuestos en (4.9), se transforman en los conjuntos

$$\begin{aligned} C &= \{(x, y) \in \Omega \mid \xi(x, y) = 0\} \\ \text{inside}(C) &= \{(x, y) \in \Omega \mid \xi(x, y) = -1\} \\ \text{outside}(C) &= \{(x, y) \in \Omega \mid \xi(x, y) = 1\} \end{aligned}$$

y la funcional resultante será

$$\begin{aligned}
 E(c_1, c_2, \chi) = & \mu \iint_{\Omega} |\nabla \xi| dx dy \\
 & + \lambda \iint_{\Omega} |u_0(x, y) - c_1|^2 \xi(x, y) dx dy \\
 & + \nu \iint_{\Omega} |u_0(x, y) - c_2|^2 (1 - \xi(x, y)) dx dy
 \end{aligned}$$

La implementación de esta función de reemplazo para la función ϕ , en el cálculo de las eigenformas, se realiza como se ilustra a continuación.

Tómese como ejemplo la figura 5.3(a), la región oscura representa valores 0 en la imagen y la región clara representa valores 1. Por tanto, los primeros son sustituidos con el valor 1 y los segundos con el valor -1. Finalmente utilizando comandos de Matlab se localiza el contorno y este es reemplazado con el valor 0, resultando la imagen de la figura 5.3(c).

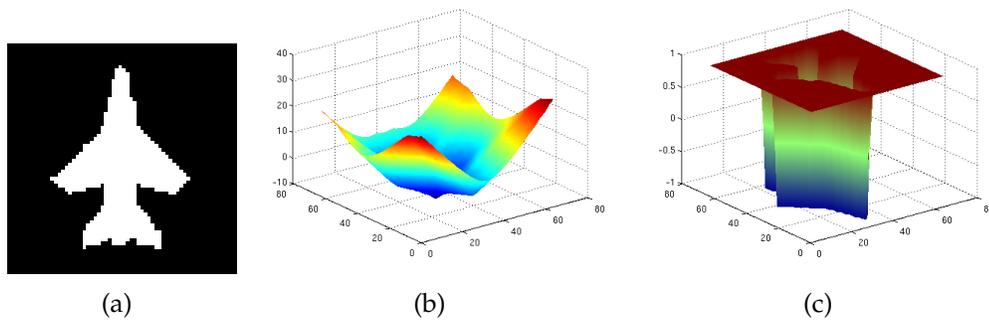


Figura 5.3: Ilustración de la función de reemplazo: (a) imagen original, (b) aplicación de la función ϕ , (c) aplicación de la función ξ .

El resultado final para cada imagen de entrada se aprecia en la figura 5.4 en la siguiente página.

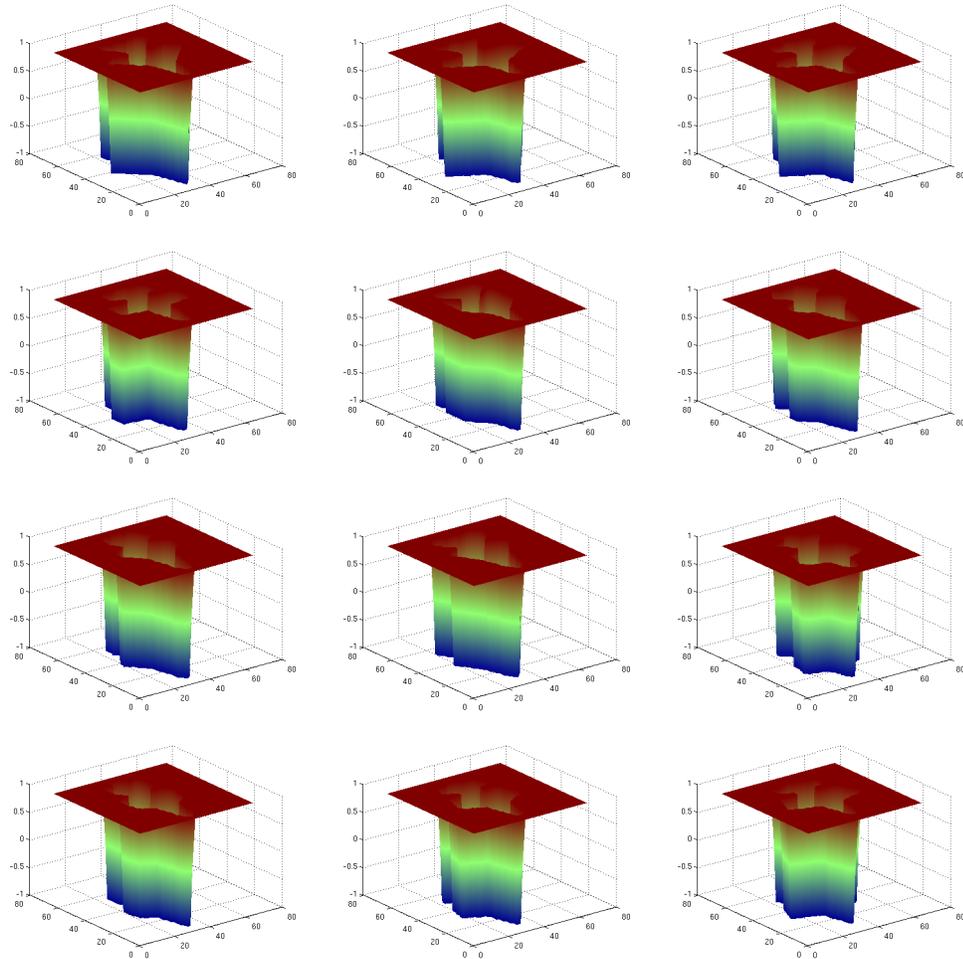


Figura 5.4: Conjunto de mapas, resultado de aplicar la función ξ , a cada imagen de entrada inicial.

Utilizando nuevamente los tres comandos de Matlab mencionados con anterioridad, los tiempos registrados utilizando la función ξ se resumen en la tabla 5.2 y finalmente una comparación de tiempos entre los diferentes métodos, se realiza en la tabla 5.3.

Function Name	Calls	Total Time	Self Time
HF	1	0.50953 s	0.01102 s
contour	12	0.49851 s	0.02388 s
newplot	12	0.24428 s	0.06058 s
cla	12	0.17956 s	0.00451 s
setdiff	24	0.12060 s	0.01525 s
setdiff>setdifflegacy	24	0.10535 s	0.05192 s
cargaImágenes	1	0.09127 s	0.00993 s
imread	12	0.06806 s	0.00822 s
ismember	24	0.05930 s	0.01003 s
addpath	1	0.05896 s	0.00649 s
legendinfo	12	0.05411 s	0.02254 s
contour>parseargs	12	0.05201 s	0.02781 s
path	1	0.04951 s	0.03910 s
imagesci/private/readpng	12	0.04768 s	0.02010 s
unique	46	0.04196 s	0.01101 s
unique>uniquelegacy	46	0.03095 s	0.03095 s
contours	12	0.02762 s	0.02245 s
findall	12	0.01698 s	0.00570 s
strcat	17	0.01328 s	0.01284 s
eigenVectors	1	0.00510 s	0.00449 s
imformats	12	0.00501 s	0.00128 s
imformats>find_in_registry	12	0.00372 s	0.00372 s
general/private/catdirs	1	0.00296 s	0.00265 s
imagesci/private/pngreadc	12	0.00242 s	0.00242 s
imagesci/private/ispng	12	0.00230 s	0.00230 s
imread>parse_inputs	12	0.00127 s	0.00127 s
fliplr	24	0.00083 s	0.00083 s
pathsep	3	0.00035 s	0.00035 s
filesep	2	0.00029 s	0.00029 s

Tabla 5.2: Nombre, número de ejecuciones, tiempo total y self time (tiempo puro) de las funciones empleadas en el cálculo de las eigenformas utilizando la función ξ .

Utilizando la función ϕ		Utilizando la función ξ	
Function Name	Total Time	Function Name	Total Time
SDF	194.94600 s	HF	0.50953 s
reinit_SD	194.94069 s	contour	0.49851 s
evolve_normal_WENO	194.67040 s	—	—
der_WENO_minus	85.17593 s	—	—
der_WENO_plus	84.26064 s	—	—
select_der_normal	14.43932 s	—	—
get_dt_normal	0.11568 s	—	—
imread	0.07222 s	imread	0.06806 s
eigenVectors	0.00631 s	eigenVectors	0.00510 s
init_normal_WENO	0.00244 s	—	—

Tabla 5.3: Comparación entre el costo computacional de la implementación del cálculo de eigenformas utilizando la función ϕ y la función ξ respectivamente.

En conclusión, la función ξ de reemplazo, mejora los tiempos significativamente de más de 3 minutos a poco más de medio segundo en total.

5.4. Adaptación de Chan-Vese

Para la adaptación del modelo de Chan-Vese, como se mencionó anteriormente, también se sustituye la función ϕ por la función ξ . Sin embargo, en esta parte del trabajo se puede observar que después de cada iteración, en la cual la imagen anterior es deformada aplicando una transformación, se producen regiones en la imagen resultante donde no existe información útil.

Para ejemplificar lo anterior, considérese la imagen de la figura 5.5(a) y obsérvese por ejemplo que el resultado de rotarla 30° , es la imagen de la figura 5.5(b) con las esquinas de color gris que representan regiones sin información útil referente a la forma anterior.

Para solucionar este problema, se puede recurrir al cálculo de la función de distancia signada o la aplicación de su reemplazo (la función ξ), ambas con objeto de recalculer la información faltante. Sin embargo, al utilizar la función ϕ , nueva-



Figura 5.5: Falta de información en una imagen trasformada

mente surge el problema del costo de tiempo, pero las pruebas realizadas con la función ξ en lugar de ϕ , en este caso no arrojan resultados correctos.

En conclusión, se recalcula la información faltante por medio de la función de distancia signada y el proceso iterativo se continua según la implementación. Por un lado se utilizan las eigenformas calculadas con la función ϕ y de igual forma las eigenformas generadas con la función ξ . El resultado para cada caso, se presenta en la figura 5.6.

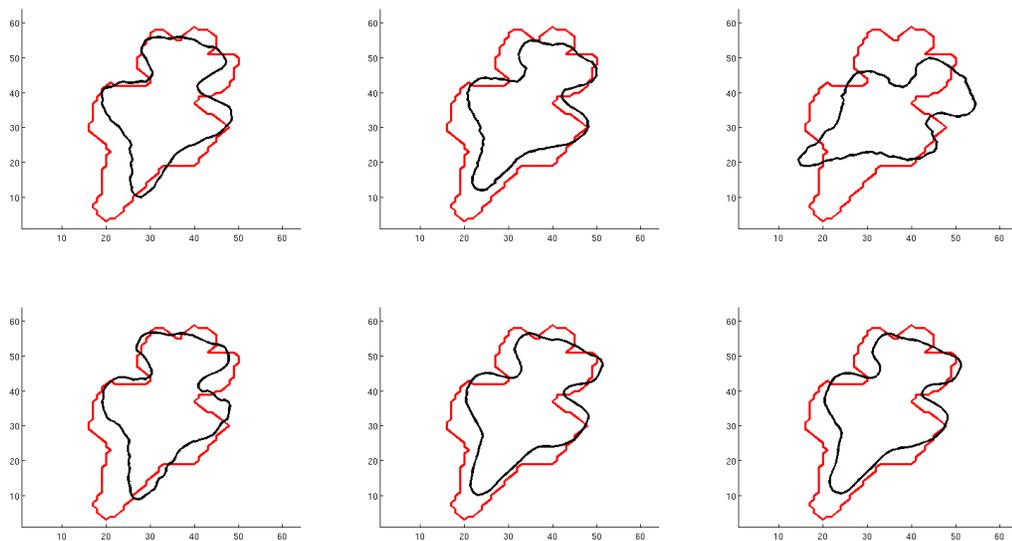


Figura 5.6: Resultado de la adaptación del modelo Chan-Vese después de 100, 200 y 400 iteraciones, utilizando las eigenformas generadas con la función ϕ (fila superior), y las eigenformas generadas con la función ξ (fila inferior).

Capítulo 6

Conclusiones

En este trabajo se presenta el problema de encontrar un método para segmentar objetos mediante el uso de su forma y de igual manera se presenta una forma de solucionar este reto. Se utilizan las ideas de los autores Tsai et al. y Mahmoodi para la creación de un algoritmo que de solución al problema planteado.

De los primeros autores se utilizan las ideas de alinear las imágenes para posteriormente poder manipularlas mejor, calcular la forma promedio y las eigenformas de cada una de las imágenes de entrada y unir todo lo anterior para generar la curva ϕ , que es utilizada en la implementación final que mediante la modificación del modelo de Chan-Vese, encuentra y por tanto segmenta la forma del objeto de interés.

En este documento, estas ideas son divididas e implementadas en las tres partes mencionadas y es en la segunda parte, es decir, en el cálculo de la curva ϕ , o más concretamente en el cálculo de las eigenformas, donde se utiliza la idea del segundo autor.

Mahmoodi sugiere que la implementación del modelo de Chan-Vese puede ser realizado sin utilizar la función de distancia signada, que durante la implementación del trabajo descrita en este documento, fue la sección que más costo en cuanto al tiempo tuvo. Por tanto la idea es reemplazar esta función mediante una que conserve las mismas propiedades del modelo de Chan-Vese, pero sin sacrificar el tiempo. Aún así, aquí se utiliza otra función llamada ξ , que es motivada por la idea anterior.

Como resultado, se presenta una tabla donde se resumen los tiempos emplea-

dos por dos algoritmos, uno con cada tipo de función y donde se puede observar que la función de reemplazo ξ , ofrece una mejoría bastante significativa.

Sin embargo, durante la implementación de la tercera parte del trabajo, se observa nuevamente la necesidad del uso de la función de distancia signada ya que cuando la imagen experimenta deformaciones, se obtienen regiones con información sin utilidad. Se puede pensar en que la solución es usar nuevamente la función de reemplazo ξ pero en esta ocasión, los resultados con esa función no son los esperados.

En conclusión la función de reemplazo funciona muy bien para la reducción de tiempos en la segunda parte del trabajo pero no en la última. Así, uno de los trabajos a futuro puede ser utilizar la idea original del segundo autor para la implementación de la tercer parte, combinándolo con las eigenformas calculadas con la función ξ y esperar mejorar aún más el tiempo junto con la obtención de resultados correctos.

Bibliografía

- [1] H.E. Abd El Munim y A.A. Farag. A shape-based segmentation approach: an improved technique using level sets. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 930–935 Vol. 2, 2005. doi: 10.1109/ICCV.2005.19.
- [2] M.S. Aslan, H. Abdelmunim, A.A. Farag, B. Arnold, E. Mostafa, y Ping Xiang. A new shape based segmentation framework using statistical and variational methods. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 717–720, 2011. doi: 10.1109/ICIP.2011.6116654.
- [3] Siciliano Bruno, Sciavicco Lorenzo, Villani Luigi, y Oriolo Giuseppe. *Robotics: Modelling, Planning and Control (Advanced Textbooks in Control and Signal Processing)*. Advanced Textbooks in Control and Signal Processing. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 978-1-84628-641-4,978-1-84628-642-1,978-1-85233-221-1.
- [4] T.F. Chan y L.A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001. ISSN 1057-7149. doi: 10.1109/83.902291.
- [5] Chih-Chung Chang y Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2: 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [6] Rajeshwar Dass y Swapna Devi. Image segmentation techniques. 2012.
- [7] C Lakshmi Devasena, R Revathí, y M Hemalatha. Video surveillance systems—a survey. *IJCSI International Journal of Computer Science Issues*, 8(4): 1694–0814.

- [8] J.E. Gentle. *Matrix Algebra: Theory, Computations, And Applications In Statistics*. Springer Texts in Statistics. Springer, 2007. ISBN 9780387708720.
- [9] G.H. Golub y C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996. ISBN 9780801854149. URL <http://books.google.es/books?id=m10a7wPX60YC>.
- [10] Michael T. Heath. *Scientific Computing: An Introductory Survey, Second Edition*. McGraw-Hill Higher Education. The McGraw-Hill Companies, Inc., 2nd edition, 1997. ISBN 0072399104,9780072399103.
- [11] Weiming Hu, Tieniu Tan, Liang Wang, y Steve Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334–352, 2004.
- [12] Ching-Kai Huang y Tsuhan Chen. Motion activated video surveillance using ti dsp. *DSPS FEST*, 99:4–6, 1999.
- [13] Cisco Systems Inc. Cisco systems ip network-centric video surveillance. pages 1–15, December 2007. URL http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6918/ps6921/ps6938/prod_white_paper0900aecd804a3e89.pdf.
- [14] Xin Jiang, Renjie Zhang, y Shengdong Nie. Image segmentation based on pdes model: A survey. In *Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference on*, pages 1–4, June 2009. doi: 10.1109/ICBBE.2009.5162922.
- [15] Wen-Xiong Kang, Qing-Qiang Yang, y Run-Peng Liang. The comparative research on image segmentation algorithms. In *Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on*, volume 2, pages 703–707, March 2009. doi: 10.1109/ETCS.2009.417.
- [16] Michael Kass, Andrew Witkin, y Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

- [17] P.J. Kelly y M.L. Weiss. *Geometry and convexity: a study in mathematical methods*. Pure and applied mathematics. Wiley, 1979. ISBN 9780471046370.
- [18] David Lyon. *Surveillance Studies: An Overview*. Wiley, 2007. ISBN 9780745635927,074563592X.
- [19] S. Mahmoodi. Shape-based active contours for fast video segmentation. *Signal Processing Letters, IEEE*, 16(10):857–860, 2009. ISSN 1070-9908. doi: 10.1109/LSP.2009.2025924.
- [20] David Mumford y Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.
- [21] K.N.N. Murthy y Y.S. Kumaraswamy. Geometric active contour detection using gradient vector flow and shape-based image segmentation. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 5, pages V5–459–V5–463, 2010. doi: 10.1109/ICACTE.2010.5579510.
- [22] Stanley Osher y Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied mathematical sciences*. Springer, 2003. ISBN 0387954821.
- [23] William H. Press, Brian P. Flannery, Saul A. Teukolsky, y William T. Vetterling. *Numerical recipes in C*. Cambridge University Press, 2 edition, 1992. ISBN 9780521431088,0521431085.
- [24] Elmer G. Rees. *Notes on Geometry (Universitext)*. Universitext (1979). U.S. Government Printing Office, 1983. ISBN 9783540120537,354012053X.
- [25] Stephen Russo. Digital video surveillance: enhancing physical security with analytic capabilities. *IBM Global Services*, pages 1–20, February 2008. URL http://www-935.ibm.com/services/us/gts/pdf/sp_wp_digital-video-surveillance.pdf.

- [26] Praylin Selva Blessy Selvaraj Assley y Helen Sulochana Chellakkon. A comparative study on medical image segmentation methods. *Applied Medical Informatics*, 34(1):31–45.
- [27] Stephen M. Smith y J. Michael Brady. Susan a new approach to low level image processing. *Int. J. Comput. Vision*, 23(1):45–78, may 1997. ISSN 0920-5691. doi: 10.1023/A:1007963824710. URL <http://dx.doi.org/10.1023/A:1007963824710>.
- [28] E. Snapper y R.J. Troyer. *Metric affine geometry*. Dover books on advanced mathematics. Dover Publications, Incorporated, 1971. ISBN 9780486661087.
- [29] Michael Spivak. *Calculus on manifolds*. Westview Press, 5 edition, 1995. ISBN 9780805390216,0805390219.
- [30] James William Thomas. *Numerical partial differential equations: finite difference methods*, volume 1 of *Graduate Texts in Mathematics*. Springer, 1995. ISBN 9780387979991.
- [31] A. Tsai, Jr. Yezzi, A., W. Wells, C. Tempany, D. Tucker, A. Fan, W.E. Grimson, y A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *Medical Imaging, IEEE Transactions on*, 22(2):137–154, 2003. ISSN 0278-0062. doi: 10.1109/TMI.2002.808355.
- [32] M. Valera y S.A. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings -*, 152(2):192–204, 2005. ISSN 1350-245X. doi: 10.1049/ip-vis:20041147.
- [33] Yanyan Wu y Yiquan Wu. Shape-based image retrieval using combining global and local shape features. In *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pages 1–5, 2009. doi: 10.1109/CISP.2009.5304693.