UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

PROGRAMA DE MAESTRIA Y DOCTORADO EN INGENIERIA

FACULTAD DE INGENIERIA

NOMBRE DE LA TESIS

Segmentación espacio-temporal de objetos en un video digital basada en un modelo con campos aleatorios de Markov

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERIA

CAMPO DE CONOCIMIENTO – INGENIERIA ELECTRICA

P R E S E N T A :

YEUDIEL VALDIVIA AGUILERA

TUTOR:

FRANCISCO JAVIER GARCÍA UGALDE

AÑO: 2008

JURADO ASIGNADO:

Presidente:	Dr. Boris Escalante Ramírez
Secretario:	Dr. Miguel Moctezuma Flores
Vocal:	Dr. Francisco Javier García Ugalde
1 er. Suplente:	Dra. Lucía Medina Gómez
2 do. Suplente:	Dra. María Elena Martínez Pérez

Lugar donde realizó la tesis:

División de Ingeniería Eléctrica de la Facultad de Ingeniería de la UNAM

TUTOR DE TESIS:

FRANCISCO JAVIER GARCIA UGALDE

_____

FIRMA

A mis padres
con especial cariño y agradecimiento

Segmentación Espacio-temporal de objetos en un video digital basada en un modelo con campos aleatorios de Markov

INTRODUCCIÓN

Introducción

La siguiente generación de estándares para la codificación de video (MPEG4), interfaces de descripción de contenido multimedia (MPEG7) y los sistemas de telecomunicaciones, no sólo necesitan incrementar la tasa de compresión, sino también presentar los datos de video de una forma bastante flexible. Esto quiere decir que los nuevos sistemas de comunicaciones tienen que hacer frente a las diferentes tareas de procesamiento de imágenes donde la segmentación es indudablemente una de las más importantes.

En años recientes la proliferación de los medios digitales ha establecido la necesidad de desarrollar herramientas para la representación eficiente, acceso y retiro de información visual. Mientras que se han propuesto varias metodologías para resolver el problema, la mayoría de los métodos más recientes confían en el análisis del contenido del medio en objetos semánticos.

La segmentación de imágenes es el proceso más importante en un gran número de aplicaciones de visión por computadora. Se trata de particionar la imagen en diferentes regiones significativas con características homogéneas usando las discontinuidades o similitudes de los componentes de la imagen. En la mayoría de los casos la segmentación de imágenes de color demuestra ser más útil que la de imágenes monocromáticas, ya que las imágenes a color expresan mucho más características de la imagen que una imagen monocromática. De hecho cada píxel se caracteriza por un gran número de combinaciones de componentes cromáticos R, G, B. Sin embargo se requieren técnicas de segmentación más complicadas para la información cromática de imágenes a color.

La segmentación de video ha sido una importante y desafiante tarea para muchas de sus aplicaciones, asume el papel principal en el contexto de la codificación basada en objetos y sus aplicaciones.

La interactividad en las actividades multimedia se está convirtiendo en una realidad. El usuario no se siente satisfecho siendo sólo un espectador pasivo y pide un papel más activo. Se han desarrollado formas sofisticadas para la

interacción directa con contenido audiovisual resultado de una evolución hacia representaciones semánticamente más trascendentes como aquellas consideradas por el estándar de codificación basada en objetos ISO MPEG-4.

En MPEG-4 una escena de video se representa como una composición de objetos de video. Se le puede permitir al usuario cambiar el *script* de composición recibido y poder eventualmente combinar los objetos recibidos con otros objetos localmente disponibles. Por otro lado, el estándar ISO MPEG-7 especifica las herramientas para crear las descripciones estandarizadas de contenido audiovisual dirigidas a una identificación eficiente y rápida, recuperación y filtrado. Estas descripciones pueden depender de una variedad de características desde aquellas que pueden ser automáticamente extraídas como textura, forma, movimiento y relaciones espaciales, hasta las de un nivel más alto como las abstractas y las que implican un valor semántico. La representación de una escena audiovisual como la composición de objetos cada uno con propiedades diferentes y un comportamiento interactivo asociado diferente asume un papel clave en las nuevas aplicaciones de multimedia y que implican soluciones de descripción y codificación audiovisual basada en objetos.

En este contexto, la segmentación de video entendida como la identificación de un conjunto de objetos (con algunas características específicas o un valor semántico) que construyen una escena de video, asume un papel principal. Los algoritmos de segmentación tienen como objetivo la identificación apropiada de objetos de acuerdo a la aplicación, además de que el análisis de la calidad de la segmentación de video resultante es de gran importancia.

La segmentación de video robusta es muy importante para las áreas de aplicación como la interacción humano-computadora, la compresión de video basada en objetos, entre otras. Para diferenciar independientemente los objetos en movimiento de una escena, una de las claves en el diseño de estos sistemas de visión es la estrategia para extraer y asociar información temporal (o movimiento) e información espacial (o intensidad) en el proceso de segmentación.

La información de movimiento es un elemento fundamental usado para la segmentación de secuencias de video. Un objeto en movimiento se caracteriza por movimiento coherente sobre su región de soporte. La escena se puede segmentar en un conjunto de regiones de tal forma que los movimientos de píxel en cada región sean consistentes con un modelo de movimiento (o una transformación paramétrica). En algunos trabajos, la información de movimiento y la segmentación son simultáneamente estimadas. Además, los diseños en capas han sido propuestos para representar múltiples objetos en movimiento en la escena con una colección de capas. Usualmente se emplea el algoritmo de la maximización de la expectativa (EM) para conocer las múltiples capas en la secuencia de la imagen.

Por otra parte, la segmentación de intensidad proporciona indicaciones importantes de las fronteras del objeto. Los métodos que combinan la segmentación de intensidad con la información de movimiento han sido propuestos en algunos trabajos. Un conjunto de regiones con pequeña variación de intensidad es provocado por una sobre-segmentación del *frame* (trama) actual. Usualmente una gráfica de adyacencia de la región o un árbol de partición se puede usar para representar las regiones en la escena. Después se forman los objetos fusionando las regiones con movimiento coherente. Los diseños de fusión de regiones presentan algunas desventajas sobre todo en las fronteras de las regiones con movimientos diferentes. Debido a que la información espacial y temporal debe interactuar a través del proceso de segmentación, se justifica que se estime simultáneamente el campo de vectores de movimiento, el campo de la segmentación de intensidad y el campo de segmentación del objeto. Los modelos gráficos proporcionan una herramienta para manejar la incertidumbre y la complejidad. En particular las redes Bayesianas y los Campos Aleatorios de Markov (CAM) están jugando un papel importante en el diseño y análisis de sistemas inteligentes incluyendo el procesamiento de imagen y video.

El objetivo de este trabajo de investigación es analizar y presentar las diferentes técnicas de segmentación de un video digital existentes en la

literatura, así como mostrar los resultados de la implementación de uno de ellos, el cual segmenta en dos capas (separando el fondo del primer plano) un video digital a color. Para ello se realizará la programación de un algoritmo en MATLAB haciendo uso de las cadenas de Markov.

Después de haber dado una introducción y justificación de este trabajo, a continuación se presenta una breve explicación de los temas cubiertos en cada uno de los capítulos.

El primer capítulo es el desarrollo del marco teórico y conceptual donde se exponen las herramientas matemáticas necesarias que se usarán a lo largo del desarrollo de la tesis.

En el segundo capítulo se presentan los principales métodos para la segmentación de imágenes estáticas.

En el Capítulo 3 se exponen los principales métodos para la estimación de movimiento, ya que un algoritmo eficiente que arroje resultados satisfactorios del campo de vectores de movimiento es esencial para una buena segmentación de video. De igual forma la segmentación basada en el movimiento juega un papel importante, por lo que algunos de estos métodos son vistos dentro del Capítulo 4.

En el Capítulo 4 también se presenta un esquema probabilístico, en el cual, la información temporal y la información espacial interactúan durante el proceso de la segmentación de video. Se propone una red Bayesiana para modelar las interacciones entre el campo de vectores de movimiento, el campo de segmentación de intensidad y el campo de la segmentación de objeto o video. Se emplea el concepto de campos aleatorios de Markov para aumentar la conectividad espacial de las regiones segmentadas. Se adopta un diseño *three-frame* para tratar el problema de las oclusiones. El criterio de segmentación es estimar el máximo de la probabilidad condicional a posteriori (Maximum a Posteriori - MAP) de las etiquetas de segmentación dadas las observaciones de tres campos dados, pertenecientes a tres *frames*

consecutivos de video. Este método de segmentación de video se compone de diseños basados en movimiento y en la fusión de regiones. Dentro de este capítulo se explica un esquema de optimización donde se propone un procedimiento que minimiza de una forma iterativa las funciones objetivas correspondientes.

En el capítulo 5 se presenta el diseño, la implementación y los resultados arrojados por un algoritmo que segmenta secuencias de video en dos capas usando haciendo uso cadenas de Markov, dicho algoritmo hace una diferencia entre el fondo (background) y el primer plano (foreground), para éste algoritmo se pueden introducir secuencias de video a color. El modelo es una fusión probabilística de las señales de movimiento, color y contraste junto con la información espacial y temporal.

CAPÍTULO 1

1. Marco Teórico

1.1. Regla de Decisión de Bayes

Se desea segmentar una secuencia de video en dos clases disjuntas de un conjunto $\Omega = \{\omega_k, k = 1,2\} = \{O, B\}$, donde *O* corresponde al objeto de interés y *B* al fondo (background).

Se considera un conjunto de N clasificadores. Cada clasificador tiene asociado un vector de características medidas para cada píxel; llamemos f_i al vector de características asociado al clasificador *i*-ésimo con i = 1,…,N. Estas características pueden ser, por ejemplo, el color, la posición dentro del cuadro, la textura del entorno del píxel, el flujo óptico, la disparidad, entre otros.

La medida de estas características en el píxel *m*-ésimo forman un patrón, $X^m = \{f_1^m, ..., f_N^m\}$.

Cada clasificador, basado en la medida de su característica asociada, es capaz de tomar una decisión respecto a cuál clase de Ω debe asignar el patrón X^m. Para tomar esta decisión cada clasificador se basa en la densidad de probabilidad condicional, $P(f_i|\omega_k)$, de la característica f_i para cada una de las clases ω_k, y en la probabilidad de ocurrencia de dichas clases, $P(\omega_k)$, llamada también probabilidad a priori. La densidad de probabilidad condicional, $P(f_i|\omega_k)$, también recibe el nombre de verosimilitud de ω_k respecto de f_i.

La regla de decisión de Bayes brinda una forma de clasificación del patrón X^m, que minimiza la probabilidad de error, utilizando las probabilidades condicionales a posteriori de cada clase dado el patrón $P(\omega_k | f_1^m, ..., f_N^m)$: se asigna X^m a la clase ω_j [1] que tiene mayor probabilidad a posteriori, esto es,

$$\text{Asignar } X^m \to \omega_j \text{ si } P(\omega_j | f_1^m, ..., f_N^m) = \max_k P(\omega_k | f_1^m, ..., f_N^m) \qquad \textbf{(1.1)}$$

En el caso particular de dos clases $\Omega = \{O, B\}$ la comparación puede escribirse como,

$$P(O | f_1^m, ..., f_N^m) \underset{X^m \to B}{\overset{X^m \to O}{\gtrless}} P(B | f_1^m, ..., f_N^m) \qquad \textbf{(1.2)}$$

[1] ω_j representa a la clase que tiene mayor probabilidad a posteriori

que se interpreta como: en caso de que se dé ">" se asigna $X^m \to O$ y en caso de que se dé "<" se asigna $X^m \to B$.

Para poder aplicar la ecuación (1.1) es necesario conocer la probabilidad a posteriori, esto se logra utilizando el *Teorema de Bayes:*

$$P\left(\omega_k \middle| f_1^m,..., f_N^m\right) = \frac{P\left(f_1^m,..., f_N^m \middle| \omega_k\right)P(\omega_k)}{P\left(f_1^m,..., f_N^m\right)} \qquad \textbf{(1.3)}$$

La verosimilitud de ω_k respecto del patrón medido es una medida de qué tan parecida es esta clase a la clase verdadera. Esta medida es pesada por la probabilidad a priori de cada clase. El denominador de la ecuación (1.3) es la densidad de probabilidad conjunta de las características, y puede calcularse mediante:

$$p\left(f_1^m,..., f_N^m\right) = \sum_{\omega_k} p\left(f_1^m,..., f_N^m \middle| \omega_k\right)P(\omega_k) = p\left(f_1^m,..., f_N^m \middle| O\right)P(O) + p\left(f_1^m,..., f_N^m \middle| B\right)P(B) \qquad \textbf{(1.4)}$$

Funcionando como un factor de escala en la ecuación (1.3); por lo tanto no es necesario calcularlo explícitamente para la comparación. Entonces, sólo es necesario el cálculo del numerador de la ecuación (1.3) para el uso de la regla de decisión de Bayes, (1.1), que queda:

$$\text{Asignar } X^m \to \omega_j \quad \text{si}$$
$$p\left(f_1^m,..., f_N^m \middle| \omega_j\right)P(\omega_j) = \max_k \left\{p\left(f_1^m,..., f_N^m \middle| \omega_k\right)P(\omega_k)\right\} \qquad \textbf{(1.5)}$$

En el caso de dos clases

$$P\left(f_1^m,..., f_N^m \middle| O\right)P(O) \underset{X^m \to B}{\overset{X^m \to O}{\gtrless}} P\left(f_1^m,..., f_N^m \middle| B\right)P(B) \qquad \textbf{(1.6)}$$

Para poder aplicar la ecuación (1.5) utilizando las medidas de todas las características, éstas deben ser utilizadas simultáneamente, es decir, se debe conocer la densidad de probabilidad conjunta $p(f_1,..., f_N|\omega_k)$ y no basta con conocer la densidad de probabilidad de cada vector de características $p\left(f_i|\omega_k\right)\forall i$ como lo hace cada clasificador individualmente.

Estimar la densidad de probabilidad de características dada la clase es independiente del número de características que se estén utilizando en teoría. En la práctica, al aumentar la dimensión del vector de características, el problema es computacionalmente mucho más costoso, además de aumentar los errores en las aproximaciones y la necesidad de mayor número de muestras para tener estimaciones confiables.

Esto nos lleva a la necesidad de realizar algunas hipótesis para poder simplificar el modelo propuesto y realizar una combinación de la información dada por las características en lo que se conoce como combinación de clasificadores o mezcla de expertos.

1.2. Combinación de clasificadores

La combinación de clasificadores permite organizar el proceso de clasificación dividiendo el problema, utilizando clasificadores más simples, creando implementaciones de soluciones en cascada o jerárquicas.

En los criterios con que los clasificadores analizan las características medidas existen dos enfoques diferentes. En el primero cada uno de los clasificadores hace uso de una característica diferente, y basado sólo en ésta genera su opinión; por ejemplo, un clasificador usa el color y otro el tamaño. En el segundo, diferentes clasificadores utilizan las mismas características medidas, y basados en diferentes criterios generan sus opiniones por ejemplo, dos clasificadores de k vecinos más cercanos (*k Nearest Neighbors*) [53] con diferente valor de k.

La combinación de clasificadores puede hacerse de tres formas diferentes. En el primer caso cada uno genera una sola etiqueta correspondiente a la clase que ha clasificado el patrón en turno. Estas etiquetas luego se combinan para dar la clasificación final. La forma en que se combinan normalmente es mediante un sistema de votación por mayoría simple. También puede darse un cierto peso a cada uno de los clasificadores teniendo en cuenta la *confianza* que se tiene en cada uno y realizar una votación ponderada.

Un segundo caso consiste en una variación del anterior, en que cada clasificador genera una lista ordenada de clases a las cuales asignar el patrón. Un sistema de votación puede ser implementado teniendo en cuenta el orden en que fueron asignadas las clases.

El tercer caso se diferencia de los dos anteriores, pues cada clasificador no devuelve una clase (o varias); la salida en este caso es la probabilidad condicional de pertenecer a cada clase (probabilidad a posteriori). La forma en que se combinan estas probabilidades puede ser mediante un promedio u otra combinación de las mismas. Dentro de estos últimos, los métodos más utilizados son la regla del producto y la regla de la suma. Existen otros esquemas de combinación a partir de estos, como la regla del máximo (maxmax), regla del mínimo (maxmin), regla de la media (maxmed).

1.2.1. Regla del producto

La regla del producto está basada en el enfoque tradicional para descomponer la densidad de probabilidad $p\left(f_1^m,...,f_N^m|\omega_k\right)$, considerando que las distintas características son estadísticamente independientes entre sí,

$$p\left(f_1^{\,m},\ldots,f_N^{\,m}\middle|\omega_k\right)=\prod_{i=1}^{N}p\left(f_i^{\,m}\middle|\omega_k\right) \qquad \textbf{(1.7)}$$

Sustituyendo la expresión anterior en la ecuación (1.5), la condición para la decisión queda:

$$P(\omega_j)\prod_{i=1}^{N}p\left(f_i^{\,m}\middle|\omega_j\right)=\max_k\left\{P(\omega_k)\prod_{i=1}^{N}p\left(f_i^{\,m}\middle|\omega_k\right)\right\} \qquad \textbf{(1.8)}$$

en función de las distribuciones a priori. Aplicando el Teorema de Bayes a $p\left(f_i^{\,m}\middle|\omega_j\right)$, podemos escribir la ecuación (1.8) en función de las probabilidades a posteriori, quedando la regla de decisión del producto,

$$\text{Asignar } X^m\rightarrow\omega_j \text{ si}$$

$$P\left(\omega_j\right)^{1-N}\prod_{i=1}^{N}P\left(\omega_j\middle|f_i^{\,m}\right)=\max_k\left\{P(\omega)^{1-N}\prod_{i=1}^{N}P\left(\omega_k\middle|f_i^{\,m}\right)\right\} \qquad \textbf{(1.9)}$$

Esta regla permite, bajo hipótesis razonables y verificables experimentalmente, simplificar el proceso de clasificación, haciendo uso de clasificadores más sencillos y el uso de vectores de características de dimensiones *manejables*. Sin embargo, si las medidas son demasiado ruidosas, puede provocar errores graves. Igualmente la hipótesis de independencia es muy fuerte y puede ser otra fuente de error en el uso de esta regla.

1.2.2. Regla de la suma

Otra regla comúnmente usada, es la regla de la suma, que puede deducirse a partir de la regla del producto, considerando una aproximación de la probabilidad a posteriori,

$$P\left(\omega_k\middle|f_i^{\,m}\right)\approx P(\omega_k)(1+\delta_{ki}) \qquad \textbf{(1.10)}$$

donde $\delta_{ki}<<1$. Sustituyendo esta expresión en la ecuación (1.8),

$$P\left(\omega_k\right)^{1-N}\prod_{i=1}^{N}P\left(\omega_k\middle|f_i^{\,m}\right)=P(\omega_k)\prod_{i=1}^{N}(1+\delta_{ki})\approx P(\omega_k)+P(\omega_k)\sum_{i=1}^{N}\delta_{ki} \qquad \textbf{(1.11)}$$

(en el último paso se expandió el producto, descartando los términos de orden mayor o igual a dos).
Por otro lado con las ecuaciones (1.10) y (1.11) se llega a:

$$\sum_{i=1}^{N}P\left(\omega_k\middle|f_i^{\,m}\right)=NP(\omega_k)+P(\omega_k)\sum_{i=1}^{N}\delta_{ki}=(N-1)P(\omega_k)+P(\omega_k)^{1-N}\prod_{i=1}^{N}P\left(\omega_k\middle|f_i^{\,m}\right) \qquad \textbf{(1.12)}$$

Entonces:

$$P(\omega_k)^{1-N}\prod_{i=1}^{N}P\left(\omega_k\left|f_i^{\,m}\right.\right)=(1-N)P(\omega_k)+\sum_{i=1}^{N}P(\omega_k\left|f_i^{\,m}\right.) \qquad \textbf{(1.13)}$$

Sustituyendo (1.13) en (1.9) resulta la regla de decisión de la suma:

$$\begin{aligned}&\text{Asignar } X^m\rightarrow\omega_j \text{ si}\\&(1-N)P(\omega_j)+\sum_{i=1}^{N}P\left(\omega_j\left|f_i^{\,m}\right.\right)\\&=\max_k\left\{(1-N)P(\omega_k)+\sum_{i=1}^{N}P(\omega_k\left|f_i^{\,m}\right.)\right\}\end{aligned} \qquad \textbf{(1.14)}$$

David M. J. Tax y otros estudiaron el problema de la clasificación utilizando la combinación de clasificadores con reglas del producto y de la suma. Sostienen que la regla de la suma obtiene mejores resultados en los casos en que las probabilidades a posteriori contienen errores en su estimación; la regla del producto mejora los resultados de la regla de la suma cuando la estimación es buena y la independencia estadística entre las características es real. Asimismo sostienen que el promediado que se realiza al aplicar esta regla reduce los errores en la estimación. Cuando las estimaciones de las probabilidades a posteriori tienen pocos errores, ambas reglas obtienen resultados similares.

Kittler y otros plantean las siguientes cotas para la regla de la suma y del producto, en caso de que las clases sean igualmente probables,

$$\prod_{i=1}^{N}P\left(\omega_k\left|f_i^{\,m}\right.\right)\le\min_{i=1}^{N}P\left(\omega_k\left|f_i^{\,m}\right.\right)\le\frac{1}{N}\sum_{i=1}^{N}P\left(\omega_k\left|f_i^{\,m}\right.\right)\le\max_{i=1}^{N}P\left(\omega_k\left|f_i^{\,m}\right.\right) \qquad \textbf{(1.15)}$$

Se puede ver que la regla de la suma ponderada con pesos iguales $\left(\frac{1}{N}\right)$, es menos estricta que la regla del producto, lo cual junto con la hipótesis que el promedio reduce los errores, permite justificar que obtenga mejores resultados cuando las probabilidades son estimadas con error.

1.3. Modelado de funciones de densidad de probabilidad

La regla de decisión de Bayes y sus variantes, planteadas en la sección 1.1 implican el conocimiento de las probabilidades a priori, $P(\omega_k)$ y las densidades de probabilidad condicionales, $p\left(f_i\left|\omega_k\right.\right)$. Estas probabilidades difícilmente son conocidas de antemano dada la estructura del problema; lo cual implica que deberán ser estimadas. Para realizar la estimación, generalmente se recurre a una muestra de datos representativos de los cuales se conocen sus características, este conjunto de muestras se conoce como conjunto de

entrenamiento o muestras de diseño, que denotaremos como $L = \{x_1,...,x_L\}$, (x_i de dimensión *d*).

De los elementos que son necesarios estimar, las probabilidades a priori de cada clase, generalmente no plantean mayores dificultades. Pero para la estimación de las funciones de densidad de probabilidad la situación es diferente. Los métodos para la estimación pueden dividirse en dos grandes categorías: paramétricos y no paramétricos. Los primeros consideran un modelo de función comúnmente gaussiano del cual estiman los parámetros que mejor se ajustan al conjunto de entrenamiento. Los métodos no paramétricos no hacen ninguna hipótesis sobre la estructura de la función. Estos métodos sirven no solo para realizar la estimación de las funciones de densidad de probabilidad, sino que también son utilizados para estimar la probabilidad a posteriori directamente, o diseñar clasificadores como por ejemplo el de K vecinos más cercanos (*k Nearest Neighbors*) [53]

1.3.1 Modelos con mezcla de gaussianas

Uno de los principales métodos paramétricos utilizados en la literatura para el modelado de funciones de densidad de probabilidad es la *Mezcla de Gaussianas* (*GMM – Gaussian Mixture Model*) [47].

El método aproxima la densidad de probabilidad por la suma de un número finito de n_G gaussianas de parámetros $\theta_i = \{\mu_i, \sum_i\}$,

$$\hat{p}(x|\Theta) = \sum_{i=1}^{n_G} \pi_i N_{\theta i}(x) \qquad \textbf{(1.16)}$$

donde π_i es la probabilidad a priori de la *i*-ésima gaussiana, Θ es el vector de incógnitas a determinar

$$\Theta = \{\pi_i,...,\pi_{n_G}, \mu_1,...,\mu_{n_G}, \sum_1,...,\sum_{n_G}\} \qquad \textbf{(1.17)}$$

y $N_{\theta i}(x)$ es un núcleo gaussiano *d*-dimensional de media μ_i y matriz de covarianza $\sum_i$

$$N_{\mu_i,\sum_i}(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\left|\sum_i\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T\sum_i^{-1}(x-\mu_i)} \qquad \textbf{(1.18)}$$

Para resolver este problema una de las posibles soluciones es aplicar el algoritmo de Maximización de la Expectativa (*Expectation Maximitation - EM)* [47], a partir de una inicialización de los parámetros Θ. Para una profundización en este algoritmo se puede consultar la referencia [47]. El número n_G de gaussianas que se utiliza en la mezcla se considera fijo en el algoritmo EM. En

el trabajo propuesto por Figueiredo y Jain [30] se propone un algoritmo no supervisado para el aprendizaje de los parámetros de una mezcla de modelos, en particular de gaussianas, determinando el número óptimo de componentes que aproximan la densidad de probabilidad junto con los parámetros de las mismas. Este algoritmo se basa en el principio de (*Minimum Message Length – MML*) para encontrar el modelo que mejor representa los datos de entrenamiento, estableciendo un compromiso entre la complejidad del modelo y la representación de los datos por éste.

Es un algoritmo computacionalmente costoso debido a que evalúa y compara las descripciones del modelo con todos los posibles valores para el número de gaussianas en la mezcla, pero a cambio obtiene la mejor forma para la descripción de los datos mediante un modelo GMM.

1.3.2. Modelos basados en núcleos

La distribución del conjunto de entrenamiento en el espacio de las muestras es un indicador de la densidad de probabilidad que se quiere estimar. Será más probable que un patrón a clasificar se encuentre en una región donde hay muchas muestras del conjunto de entrenamiento. En este concepto se basa la estimación de densidades mediante núcleos también conocido como *Ventanas de Parzen* [53].
El método consiste en sumar los aportes de los núcleos, *K*, centrados en cada uno de los puntos del conjunto de entrenamiento,

$$\hat{p}(x) = \sum_{i=1}^{L} K(x - x_i) \qquad \textbf{(1.19)}$$

El núcleo $K(\cdot)$ es una función en el espacio de muestras de volumen unidad, x y x_i son las muestras o puntos del conjunto de entrenamiento. Comúnmente el tipo de núcleo que se utiliza es un núcleo gaussiano $N_{0,\sum_K}(x)$ de media $\mu_i = 0$ y matriz de covarianza $\sum_K$.

Cuando el conjunto de entrenamiento es representativo los resultados obtenidos con este método son muy buenos. Una de las desventajas de estos métodos es que pueden implementarse con una evolución, y generalmente tiene menor carga computacional que los métodos paramétricos.

Para el modelado de algunas características, por ejemplo la posición o la forma de los objetos, GMM no es un método eficaz. GMM intenta modelar la densidad de probabilidad mediante la supervisión de núcleos elipsoides que difícilmente puedan adaptarse eficientemente a la forma de cualquier objeto. En estos casos, los métodos basados en núcleos dan mejores resultados.

1.3.3. Modelos basados en histogramas

Otro método no paramétrico utilizado se basa en histogramas como estimadores de la densidad de probabilidad. El principal inconveniente de este método es la dificultad de trabajar con altas dimensiones. Sin embargo es un método con carga computacional baja comparada con otros.

En el trabajo de Everingham y Thomas [31] se supone la independencia entre las componentes y se estima la densidad conjunta como el producto de las densidades estimadas mediante histogramas. Para esto utilizan el color y la textura como características en cada píxel. Agregan la estimación de la posición con un modelo basado en núcleos. Con este esquema realizan la comparación con el método utilizado mezcla de gaussianas para el modelado, obteniendo mejores resultados con el esquema propuesto.

CAPÍTULO 2

2.Métodos para la segmentación de imágenes

El primer paso del análisis de imágenes consiste generalmente en segmentar la imagen. La segmentación subdivide una imagen en sus partes constituyentes u objetos. El nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. Esto es, la segmentación deberá detenerse cuando los objetos de interés de una aplicación hayan sido aislados.

En general, la segmentación autónoma es una de las tareas más difíciles del procesamiento de imágenes. Esta etapa del proceso determina el eventual éxito o fracaso del análisis.

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similitud.

En la primera categoría, el método consiste en dividir una imagen basándose en los cambios bruscos de nivel de gris (detección de bordes).

Los principales métodos en la segunda categoría están basados en la umbralización, crecimiento de regiones, división y fusión de regiones, entre otros.

2.1. Métodos basados en el análisis del histograma

2.1.1. Umbralización [54]

Esta técnica está basada en un concepto muy simple: Se elige un parámetro α denominado umbral de brillo (intensidad) y entonces:

$$\text{Si} \quad A[m,n] \geq \alpha \Rightarrow B[m,n] = 1$$

$$\text{En caso contrario } B[m,n] = 0 \qquad \textbf{(2. 1)}$$

Suponiendo que A es la imagen original y B la imagen resultante o imagen segmentada y donde m,n representan el tamaño de la imagen.

Evidentemente el umbral no es único para todas las imágenes, depende del dominio y de los objetos que se quieran detectar. Existen diferentes procesos o métodos que se utilizan para obtener el umbral adecuado para una correcta segmentación:

Método *P-Tile* [55]

Este método utiliza el conocimiento a cerca del área del histograma que ocupan los objetos que se quieren detectar. Suponiendo que para una aplicación dada, los objetos ocupan sobre un *p%* del área de la imagen. Utilizando el conocimiento de esta partición (en la imagen), uno o más umbrales pueden ser elegidos asignando un porcentaje de píxeles a los objetos.

Evidentemente, este método tiene un uso muy limitado. Solamente unas pocas aplicaciones permiten estimar el área de forma general

Algoritmo *Isodata* [54]

Ésta es una técnica iterativa que se utiliza para la obtención del umbral correcto. El histograma es inicialmente segmentado en dos partes utilizando un umbral de comienzo tal como la mitad del máximo valor del rango dinámico.

A continuación se calcula la media de los valores asociados con cada una de las partes en que ha quedado segmentado el histograma m_1, m_2. Utilizando esos valores se calcula un nuevo valor umbral mediante la fórmula:

$$\alpha = \frac{(m_1 + m_2)}{2} \qquad \textbf{(2. 2)}$$

El proceso continúa hasta que en dos pasos consecutivos el valor del umbral calculado no cambia.

Algoritmo *Background-simmetry* [55]

Esta técnica asume la existencia de un pico dominante (maxp) para el background que además es simétrico con respecto a su valor máximo.

El algoritmo busca en la parte perteneciente al background (derecha del pico) el valor de intensidad que se corresponde con un cierto porcentaje de puntos (p%).

Algoritmo *Triangle* [55]

Esta técnica se basa en la detección del umbral correcto. La idea consiste en lo siguiente: Se construye una línea entre el valor máximo del histograma $b_{\max}$ y el valor más bajo $b_{\min}$. La distancia entre la línea y el histograma $h(b)$ se calcula para todos los valores de b, desde $b = b_{\min}$ hasta $b = b_{\max}$.

El valor de luminosidad b_0, donde la distancia entre $h(b_0)$ y la línea máxima es el valor umbral elegido b_0. Esta técnica es muy efectiva cuando los píxeles de los objetos producen un pico suave en el histograma.

Hasta ahora las técnicas de umbralización que se han visto tratan de obtener un valor umbral por el que se pueda dividir la imagen en dos clases bien diferenciadas.

Una técnica de *clustering* (dos o más regiones) basada en umbralizaciones del histograma es la siguiente [55]:

1. Se considera que la imagen entera es una única región y se calcula su histograma para cada componente de interés.

2. Sobre el histograma calculado se obtiene el pico mayor y se utiliza como valor umbral uno de los lados del pico. Utilizando dicho umbral se segmenta la imagen en dos regiones.

3. Se suaviza la imagen umbralizada, esta etapa se puede realizar mediante el uso de una gausiana como se indica en la sección 2.3.1.

4. Se repiten los pasos del 1-3 para cada región hasta que no se crean nuevas sub-regiones, es decir, los histogramas de cada sub-región no tienen picos significativos.

Limitaciones de los métodos basados en histogramas

Son válidos si los objetos tienen valores de intensidad constantes sobre toda la imagen pero si la iluminación no es uniforme sobre toda la escena, puede suceder que un único umbral no sea suficiente para poder segmentar la imagen.

Otra de las limitaciones de estos métodos consiste en que el histograma nos da información de la distribución global de la intensidad de una imagen. Imágenes muy diferentes pueden tener diferentes distribuciones espaciales de niveles de gris, pero tener histogramas muy similares.

2.2. Métodos de segmentación orientada a regiones

2.2.1. División de Regiones [54]

Las regiones también pueden formarse cortando la imagen original en pequeñas regiones hasta que estas verifiquen algún predicado de similitud. Sin embargo, aunque en teoría el proceso es sencillo, decidir cuando una región satisface o no el criterio de similitud, no es del todo sencillo. Un algoritmo cuya evolución fuera partir la región que no verifique el predicado de similitud en cuatro cuadrantes (aproximados), partiría la imagen en los siguientes términos:

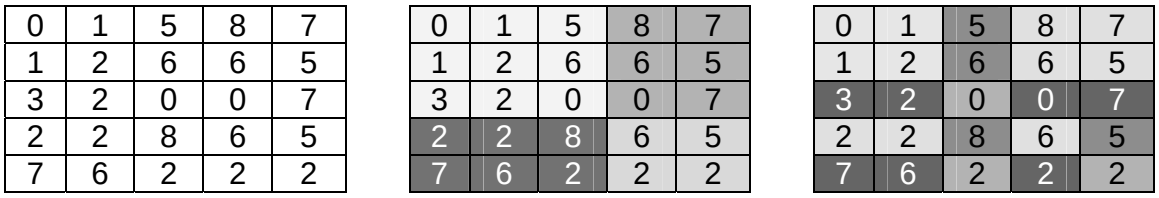

Figura 2.1

Como se puede apreciar en la figura 2.1, se corre un alto riesgo de generar una sobre-segmentación de la imagen.

2.2.2 División y Fusión [54]

Es un método que consiste en dividir la imagen en regiones uniformes. La representación que se utiliza es piramidal, una región cuadrada (m x m), en un nivel de la pirámide tiene 4 sub-regiones.

Normalmente el algoritmo comienza con la hipótesis de que la imagen completa es una única región, entonces analiza la homogeneidad de la misma (mediante un cierto criterio y propiedades). Si existe homogeneidad, la imagen se encuentra ya segmentada, si no es así, entonces la región es dividida en 4 regiones

Este proceso se repite para cada una de las regiones generadas hasta que el proceso de división no puede llevarse a cabo.

Una vez que se ha llevado a cabo el proceso de división, se comprueba para cada región generada, si es posible unirla con una región adyacente (lógicamente si satisfacen el criterio de homogeneidad establecido). El proceso termina cuando no se pueden fusionar más regiones.

2.2.3. Crecimiento de Regiones [54]

Es un método de segmentación que utiliza un principio totalmente opuesto al de división-fusión. En este método, las regiones crecen mediante agregación de píxeles similares en valor respecto a la propiedad P que se utilice para realizar la segmentación. Este tipo de algoritmos necesita que el usuario seleccione un conjunto de *puntos semilla* en la imagen. Estos puntos semilla servirán como puntos de comienzo del proceso de crecimiento de las regiones, con lo cual, el número final de regiones ha de ser como máximo igual al número de semillas sembradas por el usuario (puede ser menor, pues el algún paso del algoritmo se puede decidir unir dos regiones para formar una sola). Para poder realizar la

agregación de píxeles similares será necesario definir el concepto de similitud, que no tiene porque ser el mismo para todo tipo de aplicaciones. Posibles criterios ya utilizados en algoritmos desarrollados pueden hacer que la diferencia entre el valor del píxel a agregar y el valor de la semilla o el valor medio de la región ya formada sea menor que un cierto umbral predeterminado.

Considerar el ejemplo de la figura 2.2 donde las dos semillas introducidas están marcadas por un subrayado:

0	1	5	8	7
1	2	6	6	5
3	2	0	0	7
2	2	8	6	5
7	6	2	2	2

Figura 2.2

y considerar que se ha definido que un punto y una semilla original son similares si y sólo si su diferencia de nivel de intensidad es menor o igual que 2. Si se usa una conectividad 4, un algoritmo de crecimiento de regiones podría generar las siguientes iteraciones (a partir de cinco iteraciones ya no es posible realizar más agregaciones de píxeles).

0	1	5	8	7
1	2	6	6	5
3	**2**	0	0	7
2	2	8	6	5
7	6	2	2	2

0	1	5	8	7
1	2	6	6	5
3	**2**	0	0	7
2	2	8	6	5
7	6	2	2	2

0	1	5	8	7
1	2	6	6	5
3	**2**	0	0	7
2	2	8	6	5
7	6	2	2	2

0	1	5	8	7
1	2	6	**6**	5
3	2	0	0	7
2	2	8	6	5
7	6	2	2	2

0	1	5	8	7
1	2	6	6	5
3	**2**	0	0	7
2	2	8	6	5
7	6	2	2	2

Figura 2.3

El comienzo con un píxel semilla en particular, permitiendo que esta región crezca completamente antes de tratar otras semillas puede tener varios efectos:

- Crecimiento de regiones dominantes: Ambigüedades entorno a los bordes de regiones adyacentes pueden no ser resueltas correctamente.
- La elección de diferentes puntos semilla, puede dar lugar a diferentes segmentaciones.
- Pueden surgir problemas si un píxel semilla (elegido arbitrariamente) pertenece a un borde.

Para tratar de resolver estos problemas, se han ***desarrollado técnicas de crecimiento de regiones de forma simultánea:***

- No se permite que una única región domine completamente el proceso.
- Un cierto número de regiones crecen al mismo tiempo (regiones similares, presentan un mismo comportamiento en su crecimiento)

2.3 Métodos de Detección de Bordes [56]

Los ejes o bordes se encuentran de zonas de una imagen donde el nivel de intensidad fluctúa bruscamente, cuanto más rápido se produce el cambio de intensidad, el eje o borde es más fuerte. Un buen proceso de detección de bordes facilita la elaboración de las fronteras de objetos con lo que, el proceso de reconocimiento de objetos se simplifica.

Para poder detectar los bordes de los objetos, se deben detectar aquellos *puntos borde* que los forman. Así, un punto de borde puede ser visto como un punto en una imagen donde se produce una discontinuidad en el gradiente.

El gradiente es un vector, donde sus componentes miden la rapidez en que los valores de los píxeles cambian en la distancia y en las direcciones *x* e *y*.

dx y *dy* son las distancias en las direcciones *x* e *y* respectivamente, en términos de número de píxeles entre dos puntos.

$$\frac{\partial f(x,y)}{\partial x} = \Delta_x = \frac{f(x+d_x,y) - f(x,y)}{dx}$$
$$\frac{\partial f(x,y)}{\partial y} = \Delta_y = \frac{f(x+d_y,y) - f(x,y)}{dy} \qquad \textbf{(2. 3)}$$

Para detectar la presencia de una discontinuidad en el gradiente, debemos calcular el cambio en el gradiente en el punto (x,y). Esto se puede hacer referenciando la medida aportada por la magnitud del gradiente y su dirección.

$$M = \sqrt{\Delta^2{}_x + \Delta^2_y}$$
$$\theta = \tan^{-1}\left(\frac{\Delta_y}{\Delta_x}\right) \qquad \textbf{(2. 4)}$$

En imágenes discretas se puede considerar *dx* y *dy* en términos del número de píxeles entre dos puntos.

Para la implementación y el cálculo del gradiente se utilizan máscaras o filtros que representan o equivalen a dichas ecuaciones. En este caso, calcular el gradiente sobre toda una imagen con las condiciones de que dx=dy=1 consiste en convolucionar la imagen con unas máscaras del tipo:

$\Delta x =$ | -1 | 1 | / | 0 | 0 | $\Delta y =$ | -1 | 0 | / | 1 | 0 |

Figura 2.4

En vez de determinar el gradiente a lo largo de las direcciones x e y también podemos detectarlo en las direcciones de 45º y 135º. En este caso, las máscaras correspondientes se conocen con el nombre de ***Operadores de Roberts***

Muchas técnicas basadas en la utilización de máscaras para la detección de bordes utilizan máscaras de tamaño 3x3 o incluso más grandes.

La ventaja de utilizar máscaras grandes es que los errores producidos por efectos del ruido son reducidos mediante medias locales tomadas en los puntos en donde se superpone la máscara. Por otro lado, las máscaras normalmente tienen tamaños impares, de forma que los operadores se encuentran centrados sobre los puntos en donde se calculan los gradientes.

$$\frac{\partial f(x,y)}{\partial x} = \Delta_x = \frac{f(x+d_x,y) - f(x,y)}{dx}$$
$$\frac{\partial f(x,y)}{\partial y} = \Delta_y = \frac{f(x+d_y,y) - f(x,y)}{dy} \qquad \textbf{(2. 5)}$$

$$\Delta_x = f(x+1,y+1) - f(x,y)$$
$$\Delta_y = f(x,y+1) - f(x+1,y) \qquad \textbf{(2. 6)}$$

Otro operador muy conocido es el operador de Sobel en donde las máscaras buscan ejes en las direcciones horizontales y verticales y combinan esta información mediante la magnitud.

$\Delta x =$ | 0 | 1 |
|---|---|
| -1 | 0 |

$\Delta y =$ | 1 | 0 |
|---|---|
| 0 | -1 |

Figura 2.5

$\Delta x =$ | -1 | -2 | -1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 2 | 1 |

$\Delta y =$ | -1 | 0 | 1 |
|---|---|---|
| -2 | 0 | 2 |
| -1 | 0 | 1 |

Figura 2.6

Un operador que utiliza máscaras 3x3 y es muy parecido al de Sobel es el de Prewitt.

$\Delta x =$ | -1 | -1 | -1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

$\Delta y =$ | -1 | 0 | 1 |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |

Figura 2.7

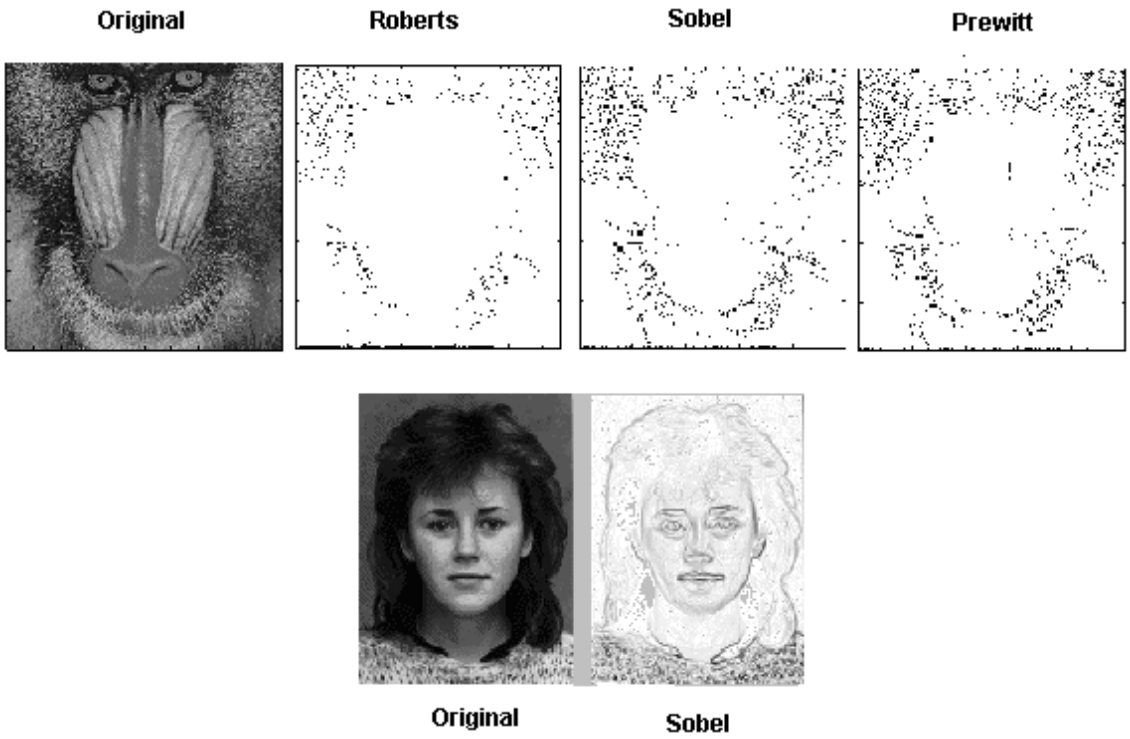

Figura 2.1. Resultados de la detección de bordes

2.3.1. Métodos basados en la 2ª derivada

Todos los operadores anteriores, tienen una aproximación hacia derivadas de primer orden sobre el valor de los píxeles en una imagen.

Existen métodos que utilizan detectores de bordes basados en derivadas de 2º orden. Uno de los más populares es el ***Operador Laplaciano*** [56].

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \qquad \textbf{(2. 7)}$$

0	-1	0
-1	4	-1
0	-1	0

Figura 2.8

Aunque el Laplaciano responde a las transiciones de intensidad, rara vez se utiliza en la práctica para la detección de bordes debido a que:

- Los operadores basados en la primera derivada son sensibles al ruido en imágenes. El Laplaciano aún lo es más.
- Genera bordes dobles.
- No existe información direccional de los ejes detectados.

Para minimizar los efectos del ruido, un método consiste en conjuntar con la operación de detección de ejes un proceso de suavizado de la imagen.

Uno de los métodos más utilizados es el suavizado por medio de una ***Gaussiana.***

- Convolucionar la imagen original con un filtro gaussiano.
- Calcular las derivas sobre la imagen suavizada.

Como ambas operaciones son lineales podemos combinar ambas operaciones de diferentes formas:

- Suavizado de la imagen y cálculo de la 2 derivada.
- Convolución de la imagen original utilizando el laplaciano del Gaussiano (***Operador LoG***).

Este método de detección de ejes fue propuesto por primera vez por Marr y Hildreth quienes introdujeron el principio de detecciones mediante el método de cruces por cero.

El principio en que se basa este método consiste en encontrar las posiciones en una imagen donde la segunda derivada toma el valor 0.

- La función gaussiana suaviza o difumina los ejes.
- La segunda derivada de la imagen difuminada es calculada. Se detectan los cruces por cero en los bordes.
- El proceso de difuminación es ventajoso ya que:
    - El laplaciano podría ser infinito en los bordes de la imagen sin suavizar.
    - La posición de los bordes se mantiene.

- El Operador LoG es también sensible al ruido, pero los efectos del ruido pueden ser reducidos si se ignoran los cruces por cero producidos por pequeños cambios en la intensidad de la imagen.
- El operador LoG nos da información de la dirección de los ejes, determinada mediante la dirección del cruce por cero.

Operador DoG [56]

Otro operador que puede aproximarse al operador LoG es el DoG. Este consiste en tomar la diferencia de dos gaussianas con diferentes desviaciones estándar. A este operador, se le conoce también como operador ***Diferencia de Gausianas*** u Operador ***de Sombrero Mexicano*** (Mexican Hat Opertator).

El operador de Marr-Hildreth [56] llegó a ser uno de los más utilizados por las siguientes razones:

- Sus fundamentos están basados en los campos receptivos de los ojos de animales.
- El operador es simétrico. Los ejes se encuentran en todas las orientaciones, cosa que no sucede con los operadores basados en la primera derivada, los cuales son direccionales.
- Los cruces por cero de la segunda derivada son más fáciles de determinar que los máximos en la primera derivada. Sólo se necesita detectar un cambio de signo en la señal. Por otro lado, los cruces por cero de una señal se encuentran siempre sobre contornos cerrados.

Problemas

- La influencia del ruido es considerable en la segunda derivada.
- La generación siempre de contornos cerrados no es realista.
- El operador DoG marca puntos considerados como ejes en algunas localizaciones donde no hay bordes.

Detector de Canny

Otro de los operadores o métodos propuestos para la detección de bordes, lo propuso J. Canny en 1993 [56].

La detección de bordes es tratada como un problema de procesado de señales y dirigida a diseñar el operador óptimo. Para ello se especificó formalmente una función objetivo que debería de ser minimizada. Dicha función fue utilizada para diseñar el operador.

La función objetivo se diseñó de forma que se obtuviese la optimización en los siguientes supuestos:

- Maximizar la relación señal ruido con objeto de obtener una buena detección. Esta maximización favorece el realce de verdaderos positivos.
- Obtener una buena localización de bordes.
- Minimizar el número de respuestas sobre bordes simples. Esto favorece la identificación de verdaderos negativos es decir, los puntos correspondientes a no-bordes no son marcados

Después de un cierto análisis, Canny determinó que la función objetivo se podía describir como la suma de 4 términos exponenciales. Al final, esta

función presenta un gran parecido a la primera derivada de una Gaussiana, así que ésta es la que se utiliza.

El procedimiento general para la detección de bordes es como sigue:

- Obtener los máximos de las derivadas parciales de la función de la imagen en las direcciones ortogonales a las direcciones de los bordes y suavizar la señal a lo largo de las direcciones de los mismos. Entonces, el operador de Canny busca los máximos en la siguiente función:

$$\frac{\partial^2}{\partial n^2}(G \otimes I) \quad n = \frac{\nabla G \otimes I}{\left|\nabla G \otimes I\right|} \qquad \textbf{(2. 8)}$$

Varios métodos se han utilizado para la realización de este proceso. Uno de ellos consiste en convolucionar la imagen con una gaussiana y buscar máximos en las derivadas parciales de la imagen transformada (utilizando máscaras parecidas a las de Sobel). El valor más alto que se produce en una cierta dirección sobre un píxel es almacenado. Así, se guardan los resultados de la convolución y la dirección del borde para cada punto.

- Cualquier valor de gradiente que no es un pico local se pone a 0. La información con respecto a la dirección de los bordes se utiliza en este proceso.
- Encontrar conjuntos de puntos de borde conectados.
- Umbralizar dichos bordes para eliminar los bordes insignificantes. Canny introduce la idea de *Proceso de Histéresis*. Se introducen dos valores de corte. Considerando un segmento de línea, si un punto presenta un valor de gradiente superior el punto de corte superior, es aceptado inmediatamente como punto de borde. Si ese valor es más pequeño que el punto de corte inferior, el punto en cuestión es desestimado. Puntos cuyo valor de gradiente se encuentra entre los puntos de corte, son considerados como bordes, si se encuentran conectados a puntos que ya han sido aceptados como puntos de bordes. Esto significa que cuando empezamos un borde, no paramos hasta que el gradiente ha descendido un valor considerable.

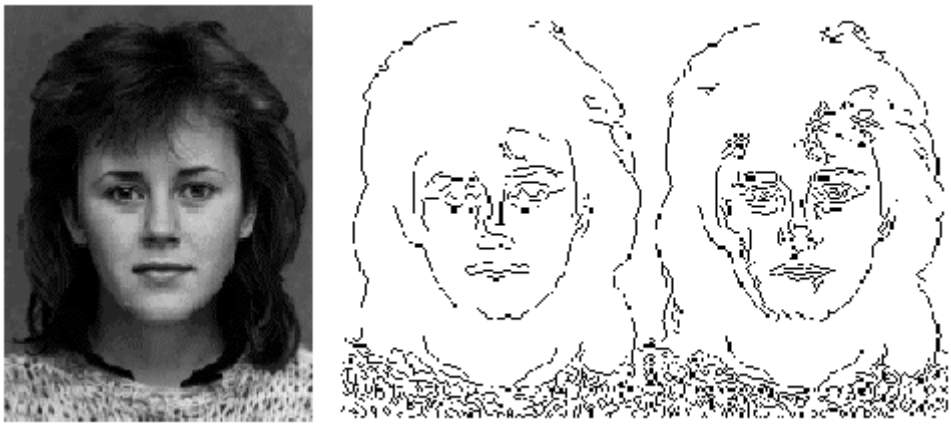

Figura 2.9. Resultados obtenidos por el operador de Canny utilizando diferentes umbrales de corte en el proceso de histéresis.

Problemas en los operadores para la detección de bordes

Existen ciertos problemas comunes en todos los operadores que utilizan el gradiente para la detección de bordes.

- Se deben de realizar elecciones en valores umbral (corte) y tamaño de las máscaras a utilizar (el tamaño condiciona el grado de suavizado, el cual puede afectar a las detecciones por cruce por cero y al máximo gradiente sobre una imagen). La posición estimada de un borde debería ser independiente del tamaño de la máscara de convolución.
- Las esquinas son a menudo omitidas a causa de que el gradiente (1D) sobre las esquinas es normalmente pequeño. Esto puede causar considerables dificultades para el etiquetado de líneas ya que éstas pueden aparecer discontinuas.

- Los operadores de primera derivada detectan solamente *step-like*. Si uno quiere encontrar líneas se necesita utilizar operadores diferentes. (por ejemplo Canny).
- Proceso diferenciales aplicados en la detección de bordes generan falsos positivos y falsos negativos.

CAPÍTULO 3

3.Métodos para la Estimación de Movimiento

La estimación de movimiento es uno de los problemas fundamentales en el procesamiento digital de video. El análisis de movimiento en secuencias de imágenes digitales tiene un gran interés para diversas aplicaciones de la visión artificial como son la robótica móvil, la monitorización de tráfico, el registrado de imágenes, la vigilancia y otras. Existen en la literatura diferentes técnicas que intentan solucionar los problemas básicos de cualquier sistema de visión que precise analizar y extraer información sobre el movimiento de los objetos de la escena. Estas técnicas se pueden agrupar en cuatro grandes categorías: las técnicas basadas en características, en flujo óptico, en modelos de movimiento y en regiones.

El desarrollo de métodos eficientes para calcular el movimiento y la estructura a partir de secuencias de imágenes es una de las áreas de mayor investigación en el campo de la visión artificial. La estimación del movimiento en imágenes tiene una amplia gama de aplicaciones, que pueden ser clasificadas dentro de tres grandes grupos:

- **Recuperación de la estructura a partir de secuencias de imágenes** (Estructura obtenida del movimiento, SFM), como son la detección de obstáculos para robots de navegación autónoma, la modelización del entorno, la adquisición automática de modelos para diseño asistido por computador (CAD), etc.

- **Compresión y reconstrucción de secuencias de imágenes** como son los casos de codificación y decodificación MPEG y la reconstrucción de imágenes afectadas por ruido.

- **Seguimiento y caracterización dinámica de objetos en movimiento** para casos como la determinación de parámetros de tráfico de automóviles, sistemas de seguridad por visión artificial y previsión meteorológica.

La recuperación del campo de movimiento parece ser la tarea esencial de cualquier sistema de visión artificial que extraiga información a partir de una secuencia de imágenes. Sin embargo, el único dato disponible es la variación espacial y temporal del patrón de brillo de la imagen. De ellas es posible obtener una aproximación del campo de movimiento denominado flujo óptico. El campo de movimiento y el flujo óptico son iguales sólo en el caso de que las variaciones espaciales del patrón de brillo correspondan a características estructurales de las superficies.

3.1. Movimiento 2-D y Movimiento aparente

Debido a que las imagenes variantes en el tiempo son proyecciones 2-D de escenas 3-D, el movimiento 2-D se refiere a la proyección del movimiento 3-D en el plano de la imagen. Se desea estimar el campo de movimiento 2-D (velocidad instantánea ó desplazamiento) de las imágenes que varían en el tiempo muestreadas en un enrejado Λ^3. Sin embargo, el campo de velocidad 2-D o campo de desplazamiento no siempre es observada por diferentes motivos. Lo que vemos puede ser lo que se conoce como campo de movimiento aparente (flujo óptico o correspondencia).

3.1.1. Movimiento 2-D

El movimiento 2-D, también conocido como *movimiento proyectado* se refiere a la perspectiva o a la proyección ortográfica del movimiento 3-D en el plano de la imagen. El movimiento 3-D se puede caracterizar en términos de la velocidad instantánea 3-D o del desplazamiento 3-D de los puntos del objeto.

El concepto del vector de desplazamiento 2-D se ilustra en la figura 3.1. Suponer que el punto **P** del objeto en el tiempo t se mueve a **P'** en el tiempo t'. La proyección de la perspectiva de los puntos **P** y **P'** en el plano de la imagen resulta en los puntos **p** y **p'**. La figura 3.2 muestra una vista 2-D del movimiento del punto **p** de la imagen en el tiempo t al punto **p'** en el tiempo t', así como la proyección de la perspectiva del movimiento 3-D de los puntos del objeto correspondientes. Debido a la operación de la proyección, todos los vectores de deslazamiento 3-D cuyos extremos estén en la línea puntada tendrán el mismo vector de desplazamiento 2-D.

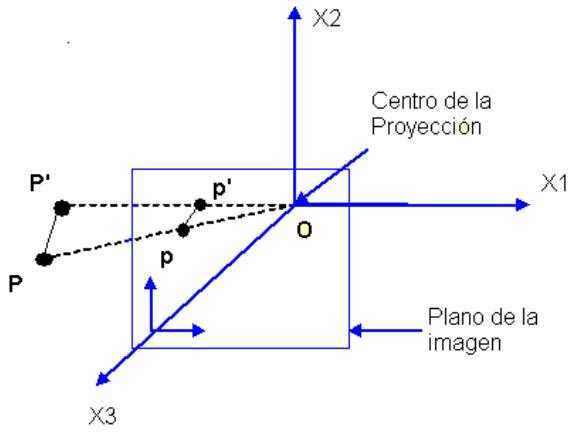

Figura 3.1. Movimiento Tri-dimensional y bi-dimensional

El desplazamiento proyectado entre los tiempos t y $t' = t + \ell\Delta t$, donde ℓ es un número entero y Δt es el intervalo de muestreo temporal, se puede definir para toda $(x,t) \in \Lambda^3$, resultando en una función del vector de desplazamiento 2-D en valor real $d_c(x,t;\ell\Delta t)$ de las variables espacio-temporales continuas.

El campo de vector de desplazamiento 2-D se refiere a una representación muestreada de esta función dada por:

$$d_p(x,t;\ell\Delta t) = d_c(x,t;\ell\Delta t), \qquad (x,t)\in\Lambda^3, \tag{3. 1}$$

O de forma equivalente,

$$d(n,k;\ell) = d_p(x,t;\ell\Delta t)\Big|_{[x_1,x_2,t]^T = V[n_1,n_2,k]^T}, \quad (n,k)\in Z^3 \tag{3. 2}$$

Donde *V* es la matriz de muestreo del enrejado Λ^3. Por lo tanto, un campo de desplazamiento 2-D es una colección de vectores de desplazamiento 2-D $d(x,t;\ell\Delta t)$, donde $(x,t)\in\Lambda^3$. La función de velocidad proyectada $V_c(x,t)$ en el tiempo t y el campo de vectores de velocidad $V_p(x,t) = v(n,k)$, para $[x_1,x_2,t]^T = V[n_1,n_2,k]^T \in \Lambda^3$ y $(n,k)\in Z^3$ puede definirse de forma similar en términos de la velocidad instantánea 3-D $(\dot{X}_1,\dot{X}_2,\dot{X}_3)$ donde el punto denota una derivada en el tiempo.

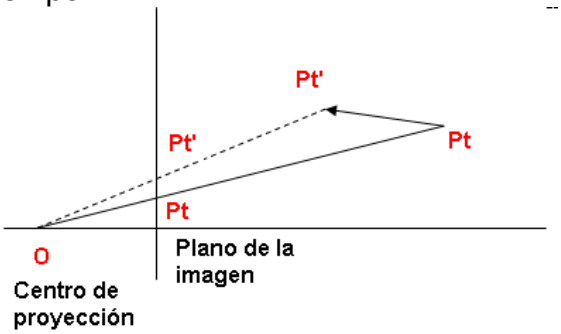

Figura 3.2. El movimiento proyectado

3.1.2. Correspondencia y Flujo Óptico

Al desplazamiento de las coordenadas x del plano de la imagen del tiempo t al tiempo t' basado en las variaciones de $s_c(x,t)$ se conoce como el vector de correspondencia. Un vector de flujo óptico se define como la tasa temporal de cambio de las coordenadas del plano de la imagen $(v_1,v_2) = (dx_1/dt, dx_2/dt)$, en un punto particular $(x,t)\in\Lambda^3$ como se determina por las variaciones espacio-temporales del patrón de intensidad $s_c(x,t)$. Esto es que corresponda al vector de velocidad instantánea de píxel. Teóricamente el flujo óptico y los vectores de correspondencia son idénticos en el límite donde $\Delta t = t'-t$ tiende a cero. En la práctica se define el campo de correspondencia (flujo óptico) como un campo de vectores del desplazamiento del píxel (velocidades) basado en las variaciones observables del patrón de intensidad de la imagen 2-D en un enrejado espacio-temporal Λ^3.

El campo de correspondencia y el campo de flujo óptico son también conocidos como el campo de *desplazamiento aparente 2-D* y campo de *velocidad aparente 2-D*, respectivamente.

El campo de correspondencia (flujo óptico) es en general, diferente del campo de desplazamiento 2-D (velocidad 2-D) debido a:

- Carencia del gradiente espacial de la imagen: Debe existir suficiente variación en el nivel de gris dentro de las regiones en movimiento para el movimiento real observado. Un ejemplo de movimiento no observable se muestra en la figura 3.3, donde un círculo con intensidad uniforme rota en su centro. Este movimiento no genera flujo óptico y por lo tanto no es observable.

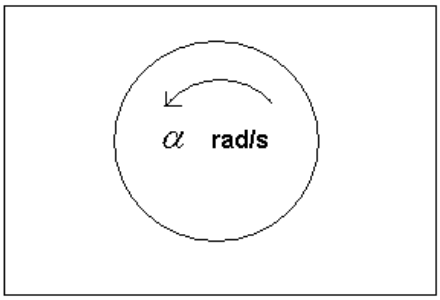

Figura 3.3. No todo el movimiento proyectado genera flujo óptico

- Cambios en la iluminación externa: Un flujo óptico observable puede no siempre corresponder al movimiento real. Por ejemplo, si la iluminación externa varía de una trama a otra como se muestra en la figura 3.4, entonces se observará un flujo óptico aunque no exista movimiento. Por lo tanto, los cambios en la iluminación externa deteriora la estimación del campo de movimiento real 2-D. En algunos casos la sombra puede variar de una trama a otra incluso si no existe cambio en la iluminación externa. Por ejemplo, si un objeto rota su superficie cambia, lo cual resulta en un cambio en la sombra. Este cambio puede causar la variación de la intensidad de los píxeles a lo largo de la trayectoria del movimiento y debe ser tomado en cuenta para la estimación de movimiento 2-D.

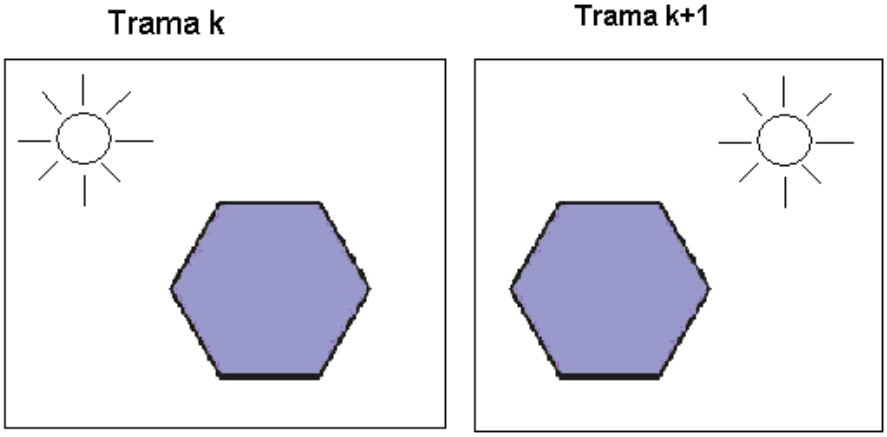

Figura 3.4. No todo el flujo óptico corresponde al movimiento proyectado

En resumen, el campo de desplazamiento 2-D y el campo de velocidad son proyecciones de sus respectivos campos 3-D en el plano de la imagen mientras que los flujos de correspondencia y flujo óptico son funciones de la velocidad y del desplazamiento percibidas del patrón de intensidad de la imagen variante en el tiempo. Ya que solo podemos observar el campo de flujo óptico y el campo de correspondencia se asume que son los mismos que en el campo de movimiento 2-D.

3.2. Estimación de movimiento 2-D

La estimación de movimiento 2-D puede ser planteada como:

1. La estimación de los vectores de correspondencia en el plano de la imagen:

$$d(x,t;\ell\Delta t) = \left[d_1(x,t;\ell\Delta t) d_2(x,t;\ell\Delta t)\right]^T \qquad \textbf{(3. 3)}$$

   entre el tiempo t y el tiempo $t+\ell\Delta t$ para todo $(x,t)\in\Lambda^3$ y ℓ es un número entero.

2. La estimación de los vectores del flujo óptico:

$$v(x,t) = \left[v_1(x,t) v_2(x,t)\right]^T \qquad \textbf{(3. 4)}$$

   para todo $(x,t)\in\Lambda^3$.

La correspondencia y los vectores del flujo óptico varían de un píxel a otro (movimiento variante en el espacio), por ejemplo, debido a la rotación de objetos en la escena y como una función del tiempo como por ejemplo debido a la aceleración de los objetos.

Problema de la correspondencia: El problema de la correspondencia puede ser visto como un problema de estimación de movimiento hace delante y hacia atrás dependiendo de si el vector de movimiento se define del tiempo t al tiempo $t+\ell\Delta t$ o del tiempo t al tiempo $t-\ell\Delta t$.

Estimación hacia delante: Dadas las muestras espacio-temporales $s_p(x,t)$ en el tiempo t y el tiempo $t+\ell\Delta t$, las cuales están relacionadas por la ecuación:

$$s_p(x_1,x_2,t) = s_p\left(x_1 + d_1(x,t;\ell\Delta t), x_2 + d_2(x,t;\ell\Delta t)\right) \qquad \textbf{(3. 5)}$$

O de forma equivalente,

$$s_k(x_1, x_2) = s_{k+\ell}\big(x_1 + d_1(x), x_2 + d_2(x)\big), \text{ tal que } t = k\Delta t \qquad \textbf{(3. 6)}$$

encontrar el vector de correspondencia de valor real $d(x) = \left[d_1(x) d_2(x)\right]^T$, donde los argumentos temporales de $d(x)$ se caen.

Estimación hacia atrás: Si definimos los vectores de correspondencia del tiempo t al tiempo $t - \ell\Delta t$, el modelo de movimiento se convierte en:

$$s_k(x_1, x_2) = s_{k-\ell}\big(x_1 + d_1(x), x_2 + d_2(x)\big), \text{ tal que } t = k\Delta t \qquad \textbf{(3. 7)}$$

Alternamente, el vector de movimiento se puede definir desde el tiempo $t - \ell\Delta t$ al tiempo *t* y se obtiene:

$$s_k(x_1, x_2) = s_{k-\ell}\big(x_1 - d_1(x), x_2 - d_2(x)\big), \text{ tal que } t = k\Delta t \qquad \textbf{(3. 8)}$$

La estimación del movimiento hacia atrás es más conveniente para la compensación de movimiento hacia delante, la cual es comúnmente utilizada en la compresión predictiva de video.

Registro de Imagen: El problema del registro es un caso especial del problema de correspondencia donde las dos tramas son globalmente cambiadas una con respecto a la otra, como por ejemplo, múltiples exposiciones de una escena estática con una cámara trasladándose.

Estimación de flujo óptico: Dadas las muestras $s_p(x_1, x_2, t)$ en un enrejado 3-D Λ^3 determinar la velocidad 2-D $\nu(x,t)$ para todo $(x,t) \in \Lambda^3$. La estimación de flujo óptico y los vectores de correspondencia de dos tramas son equivalentes, con $d(x,t;\ell\Delta t) = \nu(x,t)\ell\Delta t$, asumiendo que la velocidad permanece constante durante cada intervalo de tiempo $\ell\Delta t$. Considerar que se requieren más de dos tramas para estimar el flujo óptico en presencia de aceleración.

La estimación de movimiento 2-D, establecida como correspondencia o un problema de estimación de flujo óptico, es un problema *ill-posed* en ausencia de cualquier consideración acerca de la naturaleza del movimiento. Un problema es llamado *ill-posed* si no existe una sola solución y/o la(s) solución(es) no dependen continuamente de los datos. La estimación de movimiento 2-D tiene problemas de existencia, unicidad y continuidad:

- Existencia de una solución: No se puede establecer correspondencia para puntos cubiertos/no cubiertos del fondo. Éste es conocido como el problema de oclusión.

- Unicidad de la solución: Si los componentes del desplazamiento (o velocidad) en cada píxel son tratados como variables independientes, el número de los no conocidos es dos veces el número de las observaciones (los elementos de la diferencia de trama). Esto conduce a lo que se conoce como el problema de apertura.
- Continuidad de la solución: La estimación de movimiento es altamente sensitiva a la presencia del ruido de observación en las imágenes de video. Una pequeña cantidad de ruido puede resultar en una gran desviación en las estimaciones del movimiento.

Los problemas de oclusión y apertura son descritos en detalle a continuación.

3.2.1. Problema de Oclusión

La oclusión se refiere a la cobertura/no cobertura de una superficie debido a una rotación 3-D y a la traslación de un objeto, el cual ocupa solo una parte del campo de vista. Los conceptos de fondo cubierto y no cubierto se ilustran en la figura 3.5, donde el objeto indicado por las líneas sólidas se traslada en la dirección x_1 del tiempo *t* al tiempo *t'*. Los índices de las tramas en los tiempos *t* y *t'* son *k* y *k*+1 respectivamente. La región punteada en la trama *k* indica el fondo no cubierto por el movimiento del objeto. No existe correspondencia para estos píxeles en la trama *k*.

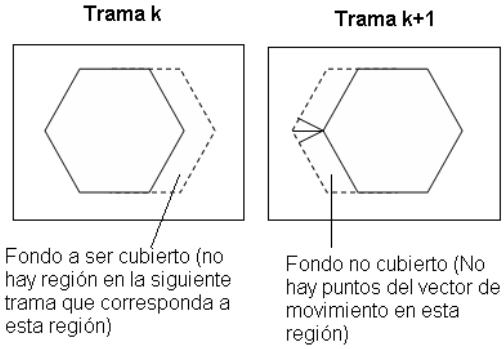

Figura 3.5. Problema de fondo cubierto/no cubierto

3.2.2. El Problema de Apertura

El problema de la apertura resulta del hecho de que la solución al problema de estimación de movimiento 2-D no es única. Si los vectores de movimiento de cada píxel se consideran como variables independientes entonces existe el doble de variables desconocidas de la ecuación (3.5). El número de ecuaciones es igual al número de píxeles en la imagen, pero para cada píxel el vector de movimiento tiene dos componentes. Un análisis teórico indica que solamente se puede determinar movimiento que es ortogonal al gradiente espacial de la imagen llamado flujo normal en cualquier píxel. El problema de apertura se indica en la figura 3.6. Supóngase que tenemos una esquina de un objeto moviéndose en la dirección x_2 hacia arriba. Si se estima el movimiento basado en una ventana local indicado por la apertura 1, entonces no es posible determinar si la imagen se mueve para arriba o perpendicular al borde.

El movimiento en la dirección perpendicular al borde se conoce como flujo normal.

Sin embargo, si observamos la apertura 2, entonces es posible estimar el movimiento correcto, ya que la imagen tiene gradiente en dos direcciones perpendiculares dentro de esta apertura. Por lo tanto es posible solucionar el problema de apertura estimando el movimiento basado en un bloque de píxeles que contiene suficiente variación de los niveles de gris. Se asume que todos estos píxeles se trasladan con el mismo vector de movimiento. Un diseño menos restrictivo para representar las variaciones de los vectores de movimiento de un píxel a otro puede obtenerse con algunos modelos de campo de movimiento 2-D paramétricos y no paramétricos.

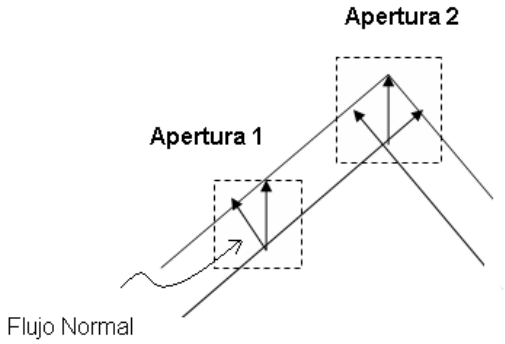

Figura 3.6. El problema de apertura

3.2.3. Modelos del campo de movimiento 2-D

Modelos paramétricos.

El objetivo de los modelos paramétricos es describir la proyección ortográfica o perspectiva del movimiento 3-D (desplazamiento o velocidad) de una superficie en el plano de la imagen. En general, los modelos paramétricos del campo de movimiento 2-D dependen de una representación de la superficie 3-D. Por ejemplo, un campo de movimiento 2-D resultado de un movimiento rígido 3-D de una superficie plana bajo la proyección ortográfica puede ser descrito por un modelo afín de 6 parámetros, mientras que bajo la proyección de perspectiva puede ser descrito por un modelo no lineal de 8 parámetros **¡Error! No se encuentra el origen de la referencia.**.

Una sub-clase de los modelos paramétricos son los llamados modelos cuasi-paramétricos los cuales tratan la profundidad de cada punto 3-D como un desconocido independiente.

Modelos no paramétricos.

La principal desventaja de los modelos paramétricos es que sólo son aplicables en caso de movimiento rígido 3-D. Alternativamente, se pueden imponer restricciones no paramétricas de uniformidad (suavidad) en el campo de movimiento 2-D sin emplear los modelos de movimiento rígido 3-D. Las restricciones no paramétricas se pueden clasificar como modelos de suavidad estocásticos o determinísticos. Algunos de los modelos no paramétricos son los descritos a continuación:

a. *Métodos basados en la Ecuación de Flujo Óptico (OFE)* [2]: Estos métodos intentan proporcionar un estimado del campo de flujo óptico en términos de gradientes de la intensidad de la imagen espacio-temporal. Con imágenes monocromáticas, se necesita usar la OFE junto con una apropiada restricción de suavidad espacio-temporal, lo cual requiere que el vector de desplazamiento varíe lentamente entre un vecindario. Si embargo en la mayoría de los casos se requiere de una restricción de suavidad (smoothness) apropiada para obtener resultados satisfactorios. Restricciones de suavidad globales provocan estimación no exacta de movimiento en las fronteras de oclusión. Restricciones de suavidad direccional más avanzadas menos discontinuidades en el campo de movimiento.

b. *Modelo de movimiento de bloque* [2]: Se asume que la imagen se compone de bloques en movimiento. Existen dos diseños para determinar el desplazamiento de los bloques de una trama a otra: los métodos de fase-correlación y los métodos *block-matching*. En los métodos de pase-correlación, el término lineal de la diferencia de fase de Fourier entre dos tramas consecutivas determina la estimación del movimiento. Los métodos de *block-matching* buscan la localidad del mejor bloque de tamaño fijo en la siguiente (y/o previo) trama(s) basado en un criterio de distancia. La forma básica de ambos métodos aplican solo para el movimiento de traslación.

c. *Métodos Pel-recursivos* [2]*:* Los métodos pel-recursivos son estimadores de desplazamiento del tipo predictor-corrector. La predicción se puede tomar como el valor del estimado del movimiento en una localidad del píxel previo o como una combinación lineal de la estimación del movimiento en una vecindad del píxel actual. La actualización se realiza mediante una minimización basada en gradiente de la diferencia de la trama desplazada (DFD – Displaced Frame Difference) en ese píxel. El paso de la predicción se considera generalmente como una restricción de suavidad implícita. Una extensión de este diseño a una estimación basada en bloque resulta en las estrategias de estimación del tipo Wiener.

d. *Métodos Bayesianos* [2]*:* Los métodos bayesianos utilizan restricciones probabilísticas de suavidad, usualmente en forma de un campo aleatorio de Gibbs para estimar el campo de desplazamiento. Su principal desventaja es la extensa cantidad de cálculos requeridos.

3.3. Métodos de Flujo Óptico

La ecuación de Flujo Óptico

Sea $s_c(x_1, x_2, t)$ una distribución de intensidad continua en el espacio del tiempo. Si la intensidad permanece constante a lo largo de la trayectoria del movimiento, se tiene que:

$$\frac{ds_c(x_1, x_2, t)}{dt} = 0 \qquad \textbf{(3. 9)}$$

donde x_1 y x_2 varían con t de acuerdo a la trayectoria del movimiento. La ecuación (3.9) es una expresión de la derivada total y denota la tasa de cambio de la intensidad a lo largo de la trayectoria del movimiento. Usando la regla de la cadena para la diferenciación, se puede expresar como:

$$\frac{\partial s_c(x;t)}{\partial x_1} v_1(x,t) + \frac{\partial s_c(x;t)}{\partial x_2} v_2(x,t) + \frac{\partial s_c(x;t)}{\partial t} = 0 \qquad \textbf{(3. 10)}$$

donde $v_1(x,t) = dx_1 / dt$ y $v_2(x,t) = dx_2 / dt$ denotan los componentes del vector de velocidad en términos de las coordenadas espaciales continuas. La expresión (3.10) se conoce como la ecuación de flujo óptico o restricción de flujo óptico.

De forma alternativa se puede expresar como:

$$\left\langle \nabla s_c(x;t), v(x,t) \right\rangle + \frac{\partial s_c(x;t)}{\partial t} = 0 \qquad \textbf{(3. 11)}$$

donde $\nabla s_c(x;t) \doteq \left[\frac{\partial s_c(x;t)}{\partial x_1} \frac{\partial s_c(x;t)}{\partial x_2} \right]^T$ y $\langle \cdot, \cdot \rangle$ denotan el producto interno del vector. La OFE dada por la ecuación (3.10) no es suficiente para especificar el campo de velocidad 2-D. La OFE produce una ecuación escalar con dos términos desconocidos $v_1(x,t)$ y $v_2(x,t)$, en cada sitio (x,t). La inspección de la ecuación (3.11) muestra que solo podemos estimar el componente del vector de flujo en la dirección del gradiente espacial de la imagen $\frac{\nabla s_c(x;t)}{\|\nabla s_c(x;t)\|}$, llamado flujo normal $v \perp (x,t)$, ya que el componente que es ortogonal al gradiente espacial de la imagen desaparece bajo el producto punto.

El flujo normal en cada sitio puede ser calculado de (3.11) como:

$$v\perp(x,t)=\frac{-\frac{\partial s_c(x;t)}{\partial t}}{\|\nabla s_c(x;t)\|} \qquad \textbf{(3. 12)}$$

La ecuación (3.10) impone una restricción en la componente del vector de flujo en la dirección del gradiente espacial de la intensidad de la imagen en cada sitio (píxel), lo cual es consistente con el problema de apertura. Nótese que los diseños con OFE requieren que la intensidad de la imagen espacio-temporal sea diferenciable además de que las derivadas parciales de la intensidad estén disponibles. En la práctica se puede mostrar que la estimación de flujo óptico de dos vistas es equivalente a la estimación de correspondencia bajo ciertas consideraciones.

Existen algunos diseños para estimar el flujo óptico de las estimaciones del flujo normal. Unos de estos diseños son los métodos diferenciales de segundo orden propuestos por H. Nagel [18] y por S. Uras [44], donde en la búsqueda de otra restricción para determinar ambas componentes del vector de flujo en cada píxel se sugiere conservar el gradiente espacial de la imagen $\nabla s_c(x;t)$, establecido como:

$$\frac{ds_c(x;t)}{dt}=0 \qquad \textbf{(3. 13)}$$

Sin embargo la restricción (3.13) no permite un cierto movimiento común como la rotación y el enfoque (zooming). Además de que las parciales de segundo orden no siempre se pueden estimar con suficiente exactitud.

Otro diseño para solucionar el problema de la apertura es asumir que el vector de movimiento permanece sin cambio sobre un bloque de píxeles en particular, denotado por β. Este método conocido como modelo de movimiento de bloque fue propuesto por B. D. Lucas y T. Kanade [4]. Aunque este método no puede manejar el movimiento rotacional es posible estimar un vector de movimiento puramente traslacional considerando que el bloque contiene suficiente variación en los niveles de gris. Con éste método la exactitud de las estimaciones del flujo depende de la exactitud de la estimación de las derivadas parciales temporales y espaciales.

El método propuesto por los investigadores en el área Horn y Schunck [5] impone una restricción de suavidad global menos estricta (restrictiva) en el campo de velocidad. Este método asume una distribución de intensidad espacio-temporal contínua.

Para la implementación en computadora todos los gradientes espaciales y temporales de la imagen necesitan ser estimados numéricamente de las muestras observadas de la imagen.

3.4. Métodos basados en bloques

La estimación y compensación de movimiento basada en bloques son de los métodos más populares. La compensación de movimiento basada en bloques ha sido adoptada en estándares internacionales para la compresión de video digital, como por ejemplo H.261 y MPEG. Aunque estos estándares no especifican un método de estimación de movimiento en particular, la estimación de movimiento basada en bloques es una elección natural. Este método es también ampliamente usado en otras aplicaciones de video digital.

Los modelos de movimiento en bloque asumen que la imagen está compuesta de bloques en movimiento. Consideraremos dos tipos de movimiento en bloques:

a. Traslación 2-D simple
b. Deformaciones 2-D de bloques

a. Movimiento traslacional de bloques [2]

La forma más simple de este modelo es la de bloques traslacionales, restringiendo el movimiento de cada bloque a solo una traslación. Por lo tanto un bloque β de tamaño N X N en la trama k centrado alrededor del píxel $n=(n_1,n_2)$ se modela como una versión cambiada globalmente de un bloque del mismo tamaño en la trama $k+\ell$ para un número entero ℓ. Esto es,

$$s(n_1,n_2,k)=s_c(x_1+d_1,x_2+d_2,t+\ell\Delta t)\Big|_{\begin{bmatrix}x\\t\end{bmatrix}=V\begin{bmatrix}n\\k\end{bmatrix}} \qquad \textbf{(3. 14)}$$

para toda $(n_1,n_2)\in\beta$, donde d_1 y d_2 son las componentes del vector de desplazamiento (traslación) para el bloque β. Recordar que el lado derecho de la ecuación (3.14) está dado en términos de la imagen continua variante en el tiempo $s_c(x_1,x_2,t)$, ya que d_1 y d_2 son valores reales. Asumiendo que los valores de d_1 y d_2 están cuantizados al número entero más cercano, el modelo (3.14) se puede simplificar como:

$$s(n_1,n_2,k)=s_c(n_1+d_1,n_2+d_2,k+\ell) \qquad \textbf{(3. 15)}$$

Observar que es posible obtener una exactitud del píxel de $1/2^L$ en la estimación del movimiento usando los métodos de fase-correlación y *block-matching* si las tramas k y $k+\ell$ en (3.15) son interpolados por un factor de L. En el modelo (3.14), los bloques β pueden no estar traslapados o traslapados. En el caso de que no estén traslapados, el bloque entero se asigna a un solo vector de movimiento. Por lo tanto se puede alcanzar la compensación de movimiento copiando píxel por píxel la información de color o de la escala de grises del bloque correspondiente en la trama $k+1$. En el caso de bloques traslapados se puede calcular el promedio de los vectores de movimiento dentro de las regiones traslapadas ó seleccionar uno de los vectores de movimiento estimados. La compensación de movimiento en el caso de bloques traslapados se discute en el trabajo propuesto por el Dr. G. Sullivan [17].

La popularidad de la compensación y estimación de movimiento basada en el modelo traslacional de bloques se origina de:

a. Pocos requerimientos generales para representar el campo de movimiento ya que se necesita un vector de movimiento por bloque.

b. Implementaciones de integración en escala muy grande (*VLSI - Very Large Scale Integration*) disponibles de bajo costo.

Sin embargo, la compensación de movimiento utilizando bloques traslacionales falla en el caso de movimiento de enfoque (zoom) y rotación, así como bajo deformaciones locales. Por otro lado, debido a que las fronteras de los objetos no coinciden con las fronteras del bloque, a bloques adyacentes se les puede asignar vectores de movimiento substancialmente diferentes.

b. Movimiento de bloque generalizado/deformable [2]

Para generalizar el modelo de bloque traslacional (3.14), notar que se puede caracterizar por una transformación (espacial) de píxel trama a trama de la forma:

$$\begin{aligned} x'_1 &= x_1 + d_1 \\ x'_2 &= x_2 + d_2 \end{aligned} \qquad \textbf{(3. 16)}$$

Donde (x'_1, x'_2) representa las coordenadas de un punto en la trama $k+l$. La transformación espacial (3.16) se puede generalizar para incluir transformaciones de coordenadas afines dadas por:

$$\begin{aligned} x'_1 &= a_1 x_1 + a_2 x_2 + d_1 \\ x'_2 &= a_3 x_1 + a_4 x_2 + d_2 \end{aligned} \qquad \textbf{(3. 17)}$$

La transformación afín (3.17) puede tratar con la rotación de bloques, así como la deformación 2-D de cuadrados (rectángulos) en paralelogramos. Otras transformaciones espaciales incluyen transformaciones de perspectiva y bilineales de coordenadas. La transformación de perspectiva está dada por:

$$x'_1 = \frac{a_1 x_1 + a_2 x_2 + a_3}{a_7 x_1 + a_8 x_2 + 1}$$
$$x'_2 = \frac{a_4 x_1 + a_5 x_2 + a_6}{a_7 x_1 + a_8 x_2 + 1} \qquad \textbf{(3. 18)}$$

Mientras que la transformación bilineal se puede expresar como:

$$x'_1 = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 + a_4$$
$$x'_2 = a_5 x_1 + a_6 x_2 + a_7 x_1 x_2 + a_8 \qquad \textbf{(3. 19)}$$

Las transformaciones de perspectiva y afín corresponden a las proyecciones de perspectiva y ortográfica del movimiento rígido 3-D de una superficie plana, respectivamente. Sin embargo, la transformación bilineal no está relacionada con ningún movimiento físico 3-D.

3.4.1. Método Fase-Correlación

Tomando la transformada de Fourier 2-D de ambos lados del modelo de movimiento discreto (3.15) con $\ell = 1$ sobre un bloque β resulta:

$$S_k(f_1, f_2) = S_{k+1}(f_1, f_2)\exp\{j2\pi(d_1 f_1 + d_2 f_2)\} \qquad \textbf{(3. 20)}$$

Donde $S_k(f_1, f_2)$ representa la transformada de Fourier 2-D de la trama k con respecto a las variables espaciales x_1 y x_2. En caso de movimiento traslacional la diferencia de las fases de Fourier 2-D de los bloques respectivos define un plano en las variables (f_1, f_2).

$$\arg\{S(f_1, f_2, k)\} - \arg\{S(f_1, f_2, k+1)\} = 2\pi(d_1 f_1 + d_2 f_2) \qquad \textbf{(3. 21)}$$

Por lo tanto el vector de movimiento entre tramas se puede estimar a partir de la orientación del plano (3.11). Existen dos problemas principales: uno es la estimación de la orientación del plano en general requiere de fase 2-D la cual no es trivial; y el otro es que no es fácil de identificar los vectores de movimiento para más de un objeto en movimiento dentro del bloque. El método de fase-correlación trata con estos problemas. Otros métodos de estimación de movimiento en el dominio de la frecuencia incluyen aquellos basados en el análisis espacio temporal 3-D en el dominio de la frecuencia usando distribuciones Weigner [24] y un conjunto de filtros de Gabor [9].

El método de fase correlación estima un cambio relativo entre dos bloques de la imagen por medias de una función de correlación cruzada (*cross-correlación*) normalizada calculada en dominio espacial 2-D de Fourier. También se basa en el principio de que un cambio relativo en el dominio espacial resulta en un término de fase lineal en el dominio de Fourier. A continuación se muestra la función de fase-correlación y después se discuten algunos aspectos referentes a su implementación.

La función de fase-correlación

La función de correlación cruzada entre las tramas *k* y *k+1* se define como:

$$c_{k,k+1}(n_1, n_2) = s(n_1, n_2, k+1) ** s(-n_1, -n_2, k) \qquad \textbf{(3. 22)}$$

Donde ** denota la operación convolución 2-D. Tomando la transformada de Fourier en ambos lados, se obtiene la expresión del espectro de potencia cruzada (*cross-power)* de valor complejo:

$$C_{k,k+1}(f_{1,} f_2) = S_{k+1}(f_1, f_2) S_k^{*}(f_1, f_2) \qquad \textbf{(3. 23)}$$

Normalizando $C_{k,k+1}(f_{1,} f_2)$ por su magnitud se obtiene la fase del espectro de potencia cruzada (*cross-power*);

$$\tilde{C}_{k,k+1}(f_1, f_2) = \frac{S_{k+1}(f_1, f_2) S_k^{*}(f_1, f_2)}{\left| S_{k+1}(f_1, f_2) S_k^{*}(f_1, f_2) \right|} \qquad \textbf{(3. 24)}$$

Considerando movimiento traslacional se sustituye la ecuación (3.20) en la ecuación (3.24) para obtener:

$$\tilde{C}_{k,k+1}(f_1, f_2) = \exp\{- j2\pi * (f_1 d_1 + f_2 d_2)\} \qquad \textbf{(3. 25)}$$

Tomando la transformada 2-D inversa de Fourier se produce la función de fase-correlación:

$$\tilde{c}_{k,k+1}(n_1, n_2) = \delta(n_1 - d_1, n_2 - d_2) \qquad \textbf{(3. 26)}$$

Se puede observar que la función de fase-correlación consiste de un impulso cuya localidad produce el vector de desplazamiento.

Consideraciones de la implementación

La implementación en la computadora del método de fase-correlación requiere remplazar las transformadas de Fourier 2-D por la DFT 2-D usando el siguiente algoritmo:

1. Calcular la DFT 2-D de los bloques respectivos de las tramas *k*th y *k*+1th.
2. Calcular la fase del espectro de potencia cruzada (cross-power) como en (3.22)
3. Calcular la DFT 2-D inversa de $\tilde{C}_{k,k+1}(f_1, f_2)$ para obtener la función fase-correlación $\tilde{c}_{k,k+1}(n_1, n_2)$.
4. Detectar la localidad del pico en la función fase-correlación.

Idealmente se espera observar un solo impulso en la función fase-correlación indicando el desplazamiento relativo entre los dos bloques. En la práctica, un número de factores contribuyen a la degeneración de la función fase-correlación para contener uno o más picos.

El tamaño del bloque es uno de los parámetros más importantes en cualquier algoritmo de estimación de movimiento basado en bloques. La selección del tamaño del bloque usualmente implica dos requerimientos opuestos. Por una parte la ventana debe ser lo suficientemente grande para ser capaz de estimar vectores grandes de desplazamiento y por otro lado debe ser lo suficientemente pequeña para que el vector de desplazamiento permanezca constante dentro de la ventana. Estos dos requerimientos contradictorios suelen tratarse con los métodos jerárquicos.

3.4.2. Método Block-Matching [2]

Este método es el más popular para la estimación de movimiento debido a la baja complejidad en *hardware*. Como resultado de lo anterior, dicho método está ampliamente disponible en VLSI y la mayoría de los codificadores H.261 y MPEG 1-2 lo utilizan para la estimación de movimiento. En la correlación de bloques (block matching), la mejor estimación del vector de movimiento se encuentra por medio de un procedimiento de búsqueda en el dominio del píxel.

La idea básica del método es determinar el desplazamiento de un píxel (n_1, n_2) en la trama *k* (trama actual) considerando un bloque $N_1 \times N_2$ centrado en (n_1, n_2) y buscar en la trama *k+1* la localidad del mejor bloque correspondiente del mismo tamaño. La búsqueda usualmente se limita a una región $N_1 + 2M_2 \times N_2 + 2M_2$ llamada ventana de búsqueda por razones computacionales, donde M_2 representa el tamaño de la ventana.
Los algoritmos de correspondencia (*matching*) de bloques difieren en:

- El criterio de correspondencia
- La estrategia de búsqueda
- La determinación del tamaño del bloque

Dentro del criterio de correspondencia, los bloques se pueden cuantificar de acuerdo a varios criterios incluyendo la correlación cruzada (*cross-correlation*) similar a la función de fase-correlación. Otros criterios son el mínimo error cuadrado medio (minimum mean square error - MSE) y la diferencia absoluta media mínima (minimum mean absolute difference - MAD).

Procedimientos de búsqueda

Para encontrar el mejor bloque de correspondencia se requiere optimizar el criterio de correspondencia sobre todos los posibles vectores de desplazamiento en cada píxel (n_1, n_2). Esto se cumple con lo que se llama *búsqueda completa,* la cual evalúa el criterio de correspondencia para todos los valores de (d_1, d_2) en cada píxel.

Una primera medida para reducir la carga computacional la búsqueda se limita usualmente a una ventana centrada en cada píxel para el cual se estimará el vector de movimiento

$$-M_1 \le d_1 \le M_1 \text{ y } -M_2 \le d_2 \le M_2$$

M_1 y M_2 son números enteros predeterminados. Otra práctica comúnmente utilizada para disminuir la carga computacional es estimar los vectores de movimiento en una rejilla de píxeles, por ejemplo, una vez cada 8 píxeles.

En la mayoría de los casos se utilizan estrategias de búsqueda más rápidas aunque éstas conducen a soluciones sub-óptimas. Algunos ejemplos de algoritmos de búsqueda más rápidos son: La búsqueda en tres pasos y la búsqueda cruzada.

Estos algoritmos evalúan la función del criterio sólo en un subconjunto predeterminado de las localidades de los vectores de movimiento. La exactitud de la estimación de movimiento en este caso depende de la aplicación.

3.4.3. Estimación jerárquica de movimiento [2]

Las representaciones jerárquicas (multi-resolución) de imágenes (tramas de una secuencia) en la forma de una pirámide Laplaciana o de una transformada ondeleta (*wavelet*) se pueden utilizar tanto con el método de fase-correlación como con el método de correlación de bloques (*block-matching*) para mejorar la estimación del movimiento. La representación piramidal de una trama muestra la resolución completa en la base y las imágenes en los niveles superiores son de resolución menor conforme se avanza hacia arriba. Dichas imágenes de menor resolución son obtenidas mediante un filtro paso bajas y sub-muestreadas.

La idea básica para el método de correspondencia de bloques o *block-matching* consiste en realizar la estimación del movimiento exitosamente en cada nivel empezando con el nivel de resolución más bajo. Los niveles de resolución más bajos sirven para determinar un cálculo aproximado del desplazamiento, utilizando bloques relativamente más grandes. Debe notarse que el tamaño relativo del bloque se puede medir como el tamaño del bloque normalizado por el tamaño de la imagen en el nivel de resolución en particular. El cálculo del vector de desplazamiento en el nivel de resolución menor se pasa al siguiente nivel de resolución mayor como una estimación inicial. Los niveles de resolución mayor sirven para afinar el cálculo del vector de desplazamiento.
En los niveles de resolución mayor se pueden utilizar ventanas de tamaño relativamente menor ya que el proceso se inicia con un cálculo inicial bueno.

3.5. Métodos Pel-recursivos [2]

Los métodos pel-recursivos son estimadores del tipo predictor-corrector de la forma:

$$\hat{d}_a(x,t;\Delta t) = \hat{d}_b(x,t;\Delta t) + u(x,t;\Delta t) \qquad \textbf{(3. 27)}$$

donde $\hat{d}_a(x,t;\Delta t)$ representa el vector de movimiento calculado en el sitio x y en el tiempo t, $\hat{d}_b(x,t;\Delta t)$ representa el cálculo del movimiento predicho y $u(x,t;\Delta t)$ es el término de actualización. Los sub-índices a y b denotan el después y el antes de la actualización en la localidad del pel (píxel) (x,t). El paso de predicción en cada píxel impone una restricción de suavidad (*smoothness*) en el cálculo y el paso de actualización fuerza la restricción de flujo óptico. El estimador (3.27) se emplea usualmente de una forma recursiva, realizando una o más iteraciones en (x,t) y después procediendo al siguiente píxel en la dirección del escaneo, de ahí el nombre de pel-recursivo.

Algunos diseños pel-recursivos se enfocan en una implementación sencilla a nivel de hardware y en la operación en tiempo real y por lo tanto emplean una predicción simple así como ecuaciones de actualización simples. Por lo general, el mejor cálculo disponible en el pel previo es tomado como el cálculo predicho para el siguiente pel seguido de una sola actualización basada en gradiente para minimizar el cuadrado de la diferencia de trama desplazada en ese pel.

Debemos remarcar que el paso de la actualización, el cual minimiza la diferencia de trama desplazada en la localidad del píxel en particular, fuerza a la ecuación (restricción) de flujo óptico en ese píxel.

3.5.1. Diferencia de trama desplazada [2]

El principio fundamental en la mayoría de los métodos para la estimación de movimiento, conocido como restricción de flujo óptico, es que la intensidad de la imagen permanece sin cambio entre una trama y otra a lo largo del movimiento (o cambia de una forma predecible o conocida). La restricción de flujo óptico debe ser empleada en la forma de la ecuación de flujo óptico como en la ecuación (3.10), o puede ser impuesta para minimizar la diferencia de trama (frame) desplazada como en los métodos pel-recursivos y *block-matching*. A continuación se presentará una descripción de la relación entre la minimización de la diferencia de trama desplazada (*Displaced Frame Difference - DFD*) y la ecuación de flujo óptico (OFE).

Considerando que la DFD entre los dos tiempos t y $t' = t + \Delta t$ está definida por:

$$dfd(x,d) \doteq s_c\left(x + d(x,t;\Delta t), t + \Delta t\right) - s_c(x,t) \qquad \textbf{(3. 28)}$$

donde $s_c(x_1, x_2, t)$ representa la distribución de la imagen variante en el tiempo, y:

$$d(x,t;\Delta t) \doteq d(x) = \left[d_1(x) \quad d_2(x)\right]^T \qquad \textbf{(3. 29)}$$

denota el campo de vectores de desplazamiento entre los tiempos t y $t + \Delta t$. Se observa que si las componentes de $d(x)$ asumen valores no enteros, se requiere de una interpolación para calcular la DFD en cada píxel y además, si $d(x)$ fuera igual al vector de desplazamiento verdadero en el sitio x sin errores de interpolación, la DFD valdría cero en el sitio bajo la restricción de flujo óptico.

Ahora, se expande el término $s_c(x + d(x), t + \Delta t)$ en series de Taylor de $(x;t)$, para $d(x)$ y Δt pequeños:

$$s_c\left(x_1 + d_1(x), x_2 + d_2(x); t + \Delta t\right) = s_c(x;t) + d_1(x)\frac{\partial s_c(x;t)}{\partial x_1} + d_2(x)\frac{\partial s_c(x;t)}{\partial x_2} + \Delta t\frac{\partial s_c(x;t)}{\partial t} + h.o.t.$$

(3. 30)

Sustituyendo (3.30) en (3.28) y despreciando los términos de más alto orden (h.o.t.) se tiene:

$$dfd(x,d) = \frac{\partial s_c(x;t)}{\partial x_1} d_1(x) + \frac{\partial s_c(x;t)}{\partial x_2} d_2(x) + \Delta t\frac{\partial s_c(x;t)}{\partial t} \qquad \textbf{(3. 31)}$$

Se investiga la relación de la DFD y la OFE en dos casos:

A. Diseños de límite 0 de Δt***:*** Se establece $dfd(x,d)=0$ y se divide ambos lados de la ecuación (3.31) por Δt y tomando cuando Δt tiende a 0, se obtiene la OFE

$$\frac{\partial s_c(x;t)}{\partial x_1}v_1(x,t)+\frac{\partial s_c(x;t)}{\partial x_2}v_2(x,t)+\frac{\partial s_c(x;t)}{\partial t}=0 \qquad \textbf{(3. 32)}$$

donde $v(x,t)=\left[v_1(x,t) \quad v_2(x,t)^T\right]$ denota la velocidad del vector en el tiempo t. Esto es, la estimación de la velocidad usando la OFE y el cálculo del desplazamiento igualando la DFD a cero o equivalente al límite donde Δt tiende a cero.

B. Para Δt ***finito****:* Un cálculo del vector de desplazamiento $\hat{d}(x)$ entre dos tramas, se puede obtener a partir de (3.31) en varias formas:

(a) Buscar $\hat{d}(x)$ igualando el lado izquierdo de la ecuación (3.31) a cero sobre un bloque de píxeles (*block matching*)

(b) Calcular $\hat{d}(x)$ igualando el lado izquierdo de la ecuación (3.31) a cero en una base de píxel a píxel utilizando un esquema de optimización basado en gradiente (pel-recursivo)

(c) Hacer $\Delta t=1$ y $dfd(x,\hat{d})=0$; resolver para $\hat{d}(x)$ usando un conjunto de ecuaciones lineales obtenidas del lado derecho de la ecuación (3.31) y usando un bloque de píxeles.

Los tres diseños anteriores son idénticos si las variaciones locales en la intensidad de la imagen espacio-temporal son lineales y si además la velocidad es constante dentro de un intervalo de tiempo Δt; esto es:

$$\hat{d}_1(x)=\hat{v}_1(x,t)\Delta t \text{ y } \hat{d}_2(x)=\hat{v}_2(x,t)\Delta t \qquad \textbf{(3. 33)}$$

En la práctica, la DFD, $dfd(x,d)$ difícilmente se vuelve exactamente cero para cualquier valor de $d(x)$ por las siguientes razones: existe ruido en la observación, existe oclusión, se introducen errores en el paso de interpolación para el caso de vectores de desplazamiento no enteros y la iluminación de la escena varía de una trama a otra. Por lo tanto, se intenta minimizar el valor absoluto de la dfd (3.28) o el lado izquierdo de la ecuación (3.31) para estimar el campo de movimiento 2-D. Los métodos pel-recursivos emplean técnicas de optimización basadas en el gradiente para minimizar el cuadrado de la dfd con una restricción de *suavidad* en el paso de predicción.

Optimización basada en el gradiente

La forma más directa de minimizar una función $f(u_1,...,u_n)$ de varios términos desconocidos es calcular la parcial respecto a cada término desconocido, igualarlas a cero y resolver simultáneamente las ecuaciones resultantes para $u_1,...,u_n$.

$$\begin{aligned} \frac{\partial f(u)}{\partial u_1} &= 0 \\ &\vdots \\ \frac{\partial f(u)}{\partial u_n} &= 0 \end{aligned} \qquad \textbf{(3. 34)}$$

Este conjunto de ecuaciones simultáneas se puede expresar como una ecuación de vector:

$$\nabla_u f(u) = 0 \qquad \textbf{(3. 35)}$$

donde ∇_u es el operador gradiente con respecto al vector desconocido u. Ya que es difícil de definir una función de criterio de forma cerrada $f(u_1,...,u_n)$ para la estimación de movimiento y/o resolver las ecuaciones (3.35) de una forma cerrada, se recurre a los métodos numéricos iterativos. Por ejemplo, la DFD es una función de las intensidades del píxel la cual no se puede expresar de una forma cerrada. Estos métodos son: el método *Steepest-Descent, Newton-Raphson y Local vs Global Minima* [2].

3.6. Métodos Bayesianos

En este caso la estimación de movimiento se formula y resuelve como un problema de estimación bayesiano. En los casos anteriores se presentaron formulaciones determinísticas del problema y se minimizó el error en la ecuación de flujo óptico y en la función de DFD. Ahora, la desviación de la DFD se modela con un proceso aleatorio que es exponencialmente distribuido. Más adelante se introduce una restricción de suavidad estocástica con el modelo del campo de vectores de movimiento 2-D en términos de una distribución de Gibbs. Los clique potenciales de una distribución Gibbsiana se seleccionan para asignar una probabilidad a priori mayor a campos de movimiento que varían lentamente. Para formular restricciones de suavidad (smoothness) direccional, campos aleatorios de Gibbs más estructurados se introducen dentro del proceso.

Ya que una estimación bayesiana requiere una optimización global se requiere conocer varios de estos métodos como son: *Simulated Annealing (SA,) Iterated Condicional Modes (ICM), Mean Field Annealing (MFA) y Highest Confidence First (HCF)* [2].

En este capítulo se de estudiaron los principales métodos para la estimación de movimiento la cual puede ser utilizada para la segmentación de un video. En el siguiente capítulo se presentarán los principales métodos para la segmentación basada en el movimiento.

CAPÍTULO 4

4. Segmentación de movimiento

Generalmente las secuencias de imágenes contienen múltiples objetos en movimiento. Los campos de flujo óptico resultante de múltiples movimientos por lo general presentan discontinuidades (bordes de movimiento). La segmentación de movimiento se refiere al etiquetado de píxeles que se asocian con cada objeto con movimiento 3-D dentro de una secuencia que presenta múltiples movimientos. Uno de los problemas es la segmentación de flujo óptico, la cual se refiere al agrupamiento de los vectores de flujo óptico que se asocian con el mismo movimiento 3-D y/o con la misma estructura. La segmentación basada en el movimiento es una parte integral de una gran variedad de problemas del análisis de secuencias de imágenes, incluyendo:

a) una mejor estimación del flujo óptimo,
b) un cálculo de la estructura y del movimiento 3-D bajo la presencia de múltiples objetos en movimiento y
c) una descripción de alto nivel de las variaciones temporales y del contenido de las imágenes de video.

En el primer caso, las etiquetas de segmentación ayudan a identificar las fronteras del flujo óptico y las regiones de oclusión. Para el segundo caso se necesita de una segmentación ya que un conjunto de parámetros distintos se requiere para modelar los vectores de flujo asociados a cada objeto con movimiento 3-D. Finalmente, en el tercer caso la información de la segmentación puede ser considerada como una descripción del alto nivel (a nivel de objeto) de la información de movimiento de trama a trama de forma opuesta a la información de movimiento de bajo nivel (a nivel de píxel) proporcionada por los vectores de flujo individuales.

Como cualquier problema de segmentación, una correcta selección de las características facilita una segmentación de movimiento efectiva. En general, la aplicación de métodos directos de flujo óptico para la segmentación de imágenes puede no producir resultados con algún significado ya que un objeto con movimiento 3-D usualmente genera una variación espacial del flujo óptico. Por ejemplo, en el caso de un solo objeto en rotación no existe flujo en el centro de rotación y la magnitud de los vectores del flujo aumenta conforme nos movemos hacia fuera del centro de rotación. Por lo anterior, en este capítulo se ha considerado un diseño basado en un modelo paramétrico para la segmentación de video basada en el movimiento donde los parámetros del modelo son las características.

Primero se presentarán los métodos directos los cuales utilizan gradientes de la imagen espacio-temporal.

4.1. Métodos Directos

En este apartado se considerarán los métodos directos para la segmentación de imágenes en regiones con movimiento independiente basada en la información de intensidad y de gradiente de una imagen espacio-temporal. Esto es en contraste a estimar primero el campo de flujo óptico entre dos tramas y después segmentar la imagen basada en el campo de flujo óptico. El primero de estos métodos es un método simple de *thresholding* (umbralización) el cual segmenta la imagen en regiones que "cambian" y regiones que "no cambian".

4.1.1. Umbralización para la detección de cambios

La umbralización se usa para segmentar una trama de video en regiones "cambiadas" y regiones "no cambiadas" con respecto a la trama previa. Las regiones que no cambian denotan el fondo estacionario y las regiones que cambian denotan el movimiento y las áreas de oclusión. Se define la diferencia de trama $FD_{k,k-1}(x_1, x_2)$ entre la trama k y la trama k-1 como:

$$FD_{k,k-1}(x_1, x_2) = s(x_1, x_2, k) - s(x_1, x_2, k-1) \qquad \textbf{(4.1)}$$

La cual es la diferencia píxel a píxel entre las dos tramas. Considerando que la iluminación permanece constante entre una trama y otra, en la localidad del píxel donde $FD_{k,k-1}(x_1, x_2)$ es diferente de cero existe una región cambiada. Sin embargo la diferencia de trama difícilmente es igual a cero debido a la presencia de ruido de observación. Para distinguir las diferencias no iguales de cero debidas al ruido de aquellas debidas a un cambio en la escena, se puede lograr una segmentación umbralizando la imagen de la diferencia como:

$$z_{k,k-1}(x_1, x_2) = \begin{cases} 1 & if \left| FD_{k,k-1}(x_1, x_2) \right| > T \\ 0 & otro \end{cases} \qquad \textbf{(4.2)}$$

donde T es un umbral apropiado. El valor T del umbral se puede elegir utilizando un algoritmo de determinación de umbral, ver apéndice B de [12]. $z_{k,k-1}(x_1, x_2)$ se conoce como el campo de etiquetas de segmentación y es igual a 1 para regiones que cambian entre una trama y otra, y es igual a 0 en otro caso. En la práctica esto puede estar aislado en la máscara de segmentación $z_{k,k-1}(x_1, x_2)$, la cual se puede eliminar realizando un post-procesamiento como por ejemplo, formar 4 u 8 regiones conectadas desechando cualquier región o regiones con menos de número predeterminado de entradas.

Hotter y Thoma [26] desarrollaron un algoritmo que utiliza parámetros de mapeo (*mapping*), dicho método se puede considerar como una estructura jerárquica de arriba hacia abajo. Empieza probando un modelo paramétrico en la región completa que cambia de una trama a otra y después divide esta región en

regiones sucesivamente más pequeñas dependiendo de que tan bien un sólo modelo se adapta a cada región o subregión. En los diseños de agrupamientos (*clustering*) y MAP (*Maximum a Posteriori*) se inicia con varias subregiones más pequeñas y se agrupan de acuerdo a algún criterio de fusión para formar segmentos. El modelo estructurado jerárquicamente se puede resumir en los siguientes pasos:

1. En el primer paso un detector de cambios inicializa la mascara de segmentación separando las regiones que cambian de las que no cambian entre una trama *k* a una trama *k+1*. Se puede utilizar un filtro morfológico para eliminar las regiones pequeñas en la mascara de detección de cambios. Cada región conectada espacialmente que cambia se interpreta como un objeto diferente.
2. Para cada objeto se calcula un modelo paramétrico diferente. La estimación de los parámetros se explica a detalle en el apartado 4.1.3.
3. La regiones que cambian que fueron encontradas en el paso 1 se divide en regiones en movimiento y para el fondo no cubierto se usan los parámetros de mapeo (*mapping)* encontrados en el paso 2. Esto se realiza de la siguiente forma: Todos los píxeles en la trama *k+1* que están en la región que cambia se trazan hacia atrás, con el inverso del vector de movimiento calculado de los parámetros de mapeo (*mapping*) encontrados en el paso 2. Si el inverso del vector de movimiento señala a un píxel en la trama *k* que esta dentro de una región que cambia, entonces el píxel en la trama *k+1* se clasifica como un píxel con movimiento, de lo contrario éste se asigna al fondo no cubierto.

Por lo tanto, la validez de los parámetros del modelo para aquellos píxeles dentro de una región en movimiento se verifica evaluando la diferencia de trama desplazada. Las regiones donde el vector de parámetro respectivo no es válido se marcan como objetos independientes para el segundo nivel jerárquico. El proceso itera entre los pasos 2 y 3 hasta que los vectores de parámetro para cada región son consistentes con la misma.

4.2. Segmentación de flujo óptico

En este caso se trata la segmentación de un campo de flujo dado usando parámetros del modelo del campo de flujo como características. Se asume que existen *K* objetos en movimiento independientes y cada vector de flujo corresponde a la proyección de un movimiento 3-D rígido de un solo objeto opaco. Por lo tanto cada movimiento distinto puede ser acertadamente descrito por un conjunto de parámetros de mapeo. Los ejemplos más comunes de modelos paramétricos, como por ejemplo, el modelo de 8 parámetros y el modelo afín, asumen implícitamente una superficie plana 3-D en movimiento.

Aproximando la superficie de un objeto real con una unión de un número pequeño de parches planos, el flujo óptico generado por un objeto real se puede

modelar por un campo de flujo cuadrático. Los vectores de flujo correspondientes a la misma superficie y el movimiento 3-D podrían tener el mismo conjunto de parámetros de mapeo (*mapping*) dentro de la misma clase.

Los métodos de segmentación basados en este modelo se pueden resumir de la siguiente forma: Primero se supone que se tiene *K* conjuntos de vectores de parámetro, donde cada conjunto define una correspondencia o un vector de flujo en cada píxel. Los vectores de flujo definidos por los parámetros de mapeo (*mapping*) se llaman vectores *basados en modelo* o vectores de flujo sintetizados. Por lo tanto se tiene *K* vectores sintetizados de flujo en cada píxel. El procedimiento de segmentación después asigna la etiqueta de cada vector sintetizado el cual está más cercano al vector de flujo calculado en cada sitio. Sin embargo, existe un pequeño problema con este esquema simple: tanto el número de clases, *K*, como los parámetros de mapeo (*mapping*) para cada clase no se conocen a priori.

Considerando un valor particular para *K*, los parámetros de mapeo para cada clase se pueden calcular en el sentido de que los vectores de flujo óptico asociados con las clases respectivas se conocen. Por lo tanto se requiere conocer los parámetros de mapeo. Esto sugiere un procedimiento iterativo similar al algoritmo de agrupamiento de *K-medias* (clustering *K-means*), donde tanto las etiquetas de segmentación como las clases no se conocen. Existen algunas variaciones de esta estrategia, una de ellas es la propuesta por G. Adiv [15].

4.2.1. Método de la transformada de Hough [2]

La transformada de Hough es una técnica de agrupamiento (*clustering*) donde las muestras de los datos “votan” por la característica más representativa en un campo característico cuantizado. En una aplicación del método de la transformada de Hough a la segmentación de flujo óptico usando el modelo de flujo afín de seis parámetros, el espacio característico de dimensión seis sería cuantificado a ciertos estados de parámetros después de que se determinan los valores máximo y mínimo de cada parámetro. Por lo tanto, cada vector de flujo $V(x) = \begin{bmatrix} v_1(x) & v_2(x) \end{bmatrix}^T$ vota por un conjunto de parámetros cuantificados que minimizan:

$$\eta^2(x) \doteq \eta_1^2(x) + \eta_2^2(x) \qquad \textbf{(4.3)}$$

Donde $\eta_1(x) = v_1(x) - a_1 - a_2 x_1 - a_3 x_2$ y $\eta_2(x) = v_2(x) - a_4 - a_5 x_1 - a_6 x_2$. Y $a_1,...,a_6$ son los seis parámetros del modelo afín.
Los conjuntos de parámetros que reciben al menos una cantidad predeterminada de votos son probables candidatos para representar el movimiento. El número de clases *K* y los conjuntos de parámetros correspondientes a ser usados en el etiquetado de los vectores de flujo se determinan de ahí.

La desventaja de éste método es la cantidad significativa de cálculos necesarios. Para mantener la carga computacional en un nivel razonable, Adiv [2] propuso un algoritmo que consta de dos etapas que involucra el procedimiento de la transformada de Hough modificada. En la primera etapa de este algoritmo, los conjuntos conectados de vectores de flujo se agrupan para formar componentes los cuales son consistentes con un solo conjunto de parámetros. Se han propuesto varias simplificaciones para facilitar los cálculos, incluyendo:

1) Descomposición del espacio de parámetros en subconjuntos $\{a_1, a_2, a_3\} \times \{a_4, a_5, a_6\}$ para realizar dos transformadas 3-D de Hough.
2) Una transformada multiresolución de Hough, donde en cada nivel de resolución el espacio de parámetros es cuantificado alrededor de las estimaciones obtenidas en el nivel previo.
3) Una técnica de Hough de multipaso, donde los vectores de flujo que son más consistentes con los parámetros, se agrupan primero.

En la segunda etapa, aquellos componentes formados en la primera etapa que son consistentes con el mismo modelo de flujo cuadrático se fusionan para formar segmentos. Se han propuesto varios criterios de fusión. En la tercera y última etapa, los vectores de flujo que no han sido agrupados se asimilan en uno de sus segmentos vecinos. En resumen, los modelos de la transformada de Hough modificada se basan en realizar primero un agrupamiento (*clustering*) de los vectores de flujo en grupos pequeños, cada uno de los cuales es consistente con el flujo generado por un movimiento de faceta plana. Después se realiza la fusión de estos grupos pequeños en segmentos mediante algún criterio de fusión *ad-hoc*.

4.2.2. Segmentación Bayesiana

El método Bayesiano busca por el máximo de la probabilidad a posteriori de las etiquetas de segmentación dados los datos de flujo óptico, la cual es una medida de qué tan bien la segmentación actual explica los datos de flujo óptico observados y de qué tan bien cumple con nuestra expectativa previa. Murray y Buxton [10] propusieron primero un método de segmentación MAP (*Maximum a Posteriori*) donde los datos de flujo óptico se modelan mediante un campo de flujo cuadrático (*piecewise*) y el campo de segmentación se modela por una distribución de Gibbs. La búsqueda de las etiquetas que maximicen la probabilidad a posteriori se realiza mediante un algoritmo conocido como SA (S*imulated Annealing*). A continuación se muestra este diseño.

Formulación del problema

Sean v_1, v_2 y z que denotan el orden lexicográfico de los componentes del vector de flujo $V(x) = [v_1(x) \quad v_2(x)]^T$ y las etiquetas de segmentación $z(x)$ en cada píxel. La función de densidad de probabilidad a posteriori (pdf) $p(z|v_1, v_2)$ del

campo de las etiquetas de la segmentación z dados los datos del flujo óptico v_1 y v_2 se puede expresar usando el teorema de Bayes como:

$$p(z|v_1, v_2) = \frac{p(v_1, v_2|z)\, p(z)}{p(v_1, v_2)} \qquad \textbf{(4.4)}$$

donde $p(v_1, v_2|z)$ es la pdf condicional de los datos del flujo óptico dada la segmentación z y $p(z)$ es la pdf a priori de la segmentación, Observar que:

1. z es un vector aleatorio de valor discreto con un espacio de muestras Ω finito
2. $p(v_1, v_2)$ es constante con respecto a las etiquetas de segmentación, por lo que se pueden ignorar para propósitos de segmentación.

La estimación MAP después maximiza el numerador de la ecuación (4.4) sobre todas las posibles realizaciones del campo de segmentación $z = \omega, \omega \in \Omega$.
La probabilidad condicional $p(v_1, v_2|z)$ es una medida de que tan bien el modelo de flujo cuadrático, donde los parámetros del modelo $a_1,...,a_8$ dependen de la etiqueta de segmentación, se ajusta al campo de flujo óptico calculado v_1 y v_2

$$\begin{aligned} v_1 &= a_1 + a_2 x_1 + a_3 x_2 + a_7 x_1^2 + a_8 x_1 x_2 \\ v_2 &= a_4 + a_5 x_1 + a_6 x_2 + a_7 x_1 x_2 + a_8 x_2^2 \end{aligned} \qquad \textbf{(4.5)}$$

Considerando una diferencia entre el flujo observado $v(x)$ y el flujo sintetizado, la ecuación (4.6) se modela con ruido blanco, gaussiano con media cero y varianza σ^2.

$$\begin{aligned} \tilde{v}_1(x) &= a_1 x_1 + a_2 x_2 - a_3 + a_7 x_1^2 + a_8 x_1 x_2 \\ \tilde{v}_2(x) &= a_4 x_1 + a_5 x_2 - a_6 + a_7 x_1 x_2 + a_8 x_2^2 \end{aligned} \qquad \textbf{(4.6)}$$

La pdf condicional del campo de flujo óptico dadas las etiquetas de segmentación se puede expresar como:

$$p(v_1, v_2|z) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left\{ -\sum_{i=1}^{M} \eta^2(x_i) / 2\sigma^2 \right\} \qquad \textbf{(4.7)}$$

Donde M es el número de vectores de flujo disponibles en los sitios x_i, y $\eta^2(x_i)$ es la desviación normal cuadrada de los vectores reales del flujo el cual se predice del modelo de flujo cuadrático, $\eta^2(x_i)$ está dado por la siguiente ecuación:

$$\eta^2(x_i) = \left(v_1(x_i) - \tilde{v}_1(x_i)\right)^2 + \left(v_2(x_i) - \tilde{v}_2(x_i)\right)^2 \qquad \textbf{(4.8)}$$

Considerando que el modelo de flujo cuadrático es más o menos exacto, esta desviación se deba a los errores en la segmentación y al ruido de observación. La pdf a priori se modela por una distribución de Gibbs la cual introduce de forma efectiva las restricciones locales en la interpretación (segmentación) y esta dada por:

$$p(z) = \frac{1}{Q}\sum_{\omega\in\Omega}\exp\{-U(z)\}\delta(z-\omega) \qquad \textbf{(4. 9)}$$

donde Ω denota el espacio de muestras discretas de z, Q es la función de partición y se define por la siguiente ecuación:

$$Q = \sum_{\omega\in\Omega}\exp\{-U(\omega)\} \qquad \textbf{(4.10)}$$

Y $U(\omega)$ es la función potencial la cual se puede expresar como la suma de los clique potenciales locales $V_C\left(z(x_i), z(x_j)\right)$. Las restricciones *a priori* en la estructura de las etiquetas de la segmentación se pueden expresar en términos de las funciones de los clique potenciales. Sustituyendo las ecuaciones (4.7) y (4.9) en la ecuación (4.4) y tomando el logaritmo de la expresión resultante, la maximización de la distribución de la probabilidad *a posteriori* se puede realizar minimizando la función de costo:

$$E = \frac{1}{2\sigma^2}\sum_{i=1}^{M}\eta^2(x_i) + U(\omega) \qquad \textbf{(4.11)}$$

El primer término describe qué tan bien los datos predichos concuerdan con las mediciones del flujo óptico real y el segundo término mide qué tanto cumple la segmentación con nuestras expectativas.

El Algoritmo

Ya que los parámetros del modelo correspondientes a cada píxel no se conocen a priori, la segmentación MAP alterna entre la estimación de los parámetros del modelo y la asignación de las etiquetas de la segmentación para optimizar la función de costo (4.11) basada en un procedimiento de SA (*Simulated Annealing*)[1]. Dado el campo de flujo v y el número de los movimientos independientes *K*, la segmentación MAP usando el algoritmo *Metrópolis* se pude resumir en los siguientes pasos:

[1] El nombre procede de la técnica de generación de aleaciones. Son decisiones en función de la temperatura del sistema. Se usa ampliamente en problemas reales donde se requiere un costo computacional bajo. A temperaturas altas los átomos tienen alta energía y mayor libertad para situarse en la red. La estructura regular final corresponde a un estado de mínima energía [57].

1. Iniciar con un etiquetado *z* de los vectores de flujo óptico. Calcular los parámetros de mapeo (*mapping*) $a = [a_1 ... a_8]^T$ para cada región. Fijar la temperatura inicial para el SA.
2. Escanear los sitios de píxel de acuerdo a una convención predefinida. En cada sitio x_i:

(a) Perturbar de forma aleatoria la etiqueta $z_i = z(x_i)$

(b) Decidir si aceptar o rechazar esta perturbación basándose en el cambio de ΔE en la función de costo (4.11),

$$\Delta E = \frac{1}{2\sigma^2}\Delta\eta^2(x_i) + \sum_{x_j \in Nx_i} \Delta V_C\left(z(x_i), z(x_j)\right) \qquad \textbf{(4.12)}$$

donde Nx_i denota el vecindario del sitio x_i y $V_C\left(z(x_i), z(x_j)\right)$ está dado por la siguiente ecuación:

$$V_C\left(z(x_i), z(x_j)\right) = \begin{cases} -\beta & if \quad z(x_i) = z(x_j) \\ +\beta & otro \quad caso \end{cases} \qquad \textbf{(4.13)}$$

donde β es un número positivo.

El primer término de la ecuación (4.12) indica si la etiqueta perturbada es más consistente con el campo de flujo dado, determinado por la ecuación (4.8) y el segundo término de (4.12) indica si éste esta de acuerdo con el modelo del campo de segmentación previo.

3. Después de que todos los sitios de píxel son visitados una vez, se vuelven a estimar los parámetros de mapeo (*mapping*) para cada región basado en la nueva configuración de las etiquetas de segmentación.
4. Salir si el criterio se satisface. En caso contrario, bajar la temperatura y regresar al paso 2.

Los métodos de segmentación de movimiento presentados en este capítulo están limitados por la exactitud de los cálculos del flujo óptico. Para estimaciones del flujo óptico más exactas, se obtendrán mejores resultados en la segmentación. A continuación se muestra un diseño donde tanto la estimación del flujo óptico como la segmentación interactúan mutuamente.

4.3. Estimación de movimiento y segmentación simultáneas

Como ya se mencionó unos resultados satisfactorios de la segmentación de flujo óptico dependen de la exactitud con que se estime el campo de flujo óptico y viceversa. De lo anterior, que el cálculo tanto del flujo óptico como de la

segmentación tienen que realizarse de forma simultánea. En esta sección se presenta un diseño bayesiano simultáneo basado en una representación del campo de movimiento como la suma de un campo paramétrico y un campo residual. La interdependencia del campo de segmentación y del campo de flujo óptico se expresa en términos de una distribución de Gibbs dentro de un diseño MAP. Encontrar las estimaciones de un conjunto denso de vectores de movimiento y de un conjunto grande de etiquetas de segmentación así como un conjunto de parámetros de mapeo requiere de un proceso de optimización el cual se puede realizar mediante un algoritmo HCF *(Highest Confidence First)* [59] ó un algoritmo ICM *(Iterated Condicional Modes)* [58].

La mayoría de los diseños que realizan una estimación simultánea del campo de flujo óptico y del campo de la segmentación se basan en los principios presentados en este modelo.

4.3.1 Modelo del Campo de Movimiento

Se supone que en una escena existen *K* objetos opacos con movimientos independientes, donde el movimiento inducido 2-D por cada objeto se puede aproximar con un modelo paramétrico como en (4.6) o por un modelo afín de 6 parámetros. Después, el campo de flujo óptico $v(x)$ se puede representar como la suma de un campo de flujo paramétrico $\tilde{v}(x)$ y un flujo residual no paramétrico $v_r(x)$, esto es:

$$v(x) = \tilde{v}(x) + v_r(x) \qquad \textbf{(4.14)}$$

Las componentes paramétricas del campo de movimiento dependen de la etiqueta de segmentación $z(x)$, la cual puede tomar los valores de 1,…, *K*.

4.3.2 Formulación del Problema

El modelo basado en MAP (*Maximum a Posteriori*) simultáneo consiste en maximizar la pdf a posteriori (ec. 4.15) con respecto al flujo óptico v_1, v_2 y a las etiquetas de segmentación z. A través de un buen modelado de estas funciones de densidad de probabilidad, se puede expresar un conjunto de restricciones que ayuden a mejorar las estimaciones.

$$p(v_1, v_2, z | g_k, g_{k+1}) = \frac{p(g_{k+1} | g_k, v_1, v_2, z)\, p(v_1, v_2 | z, g_k)\, p(z | g_k)}{p(g_{k+1} | g_k)} \qquad \textbf{(4.15)}$$

La primera pdf condicional $p(g_{k+1} | g_k, v_1, v_2, z)$ proporciona una medida de qué tan bien las estimaciones del desplazamiento y de la segmentación concuerdan con la trama *k+1* dada la trama *k* y se modela con una distribución de Gibbs como:

$$p(g_{k+1}|g_k, v_1, v_2, z) = \frac{1}{Q_1}\exp\{-U_1(g_{k+1}|g_k, v_1, v_2, z)\} \quad \textbf{(4.16)}$$

donde Q_1 es la función de partición la cual es constante y $U_1(g_{k+1}|g_k, v_1, v_2, z)$ está dado por la ecuación (4.17) y se le conoce como el potencial de Gibbs.

$$U_1(g_{k+1}|g_k, v_1, v_2, z) = \sum_x [g_k(x) - g_{k+1}(x + v(x)\Delta t)]^2 \quad \textbf{(4. 17)}$$

En este caso, el potencial de Gibbs corresponde al cuadrado normal de la diferencia de trama desplazada (DFD de sus siglas en inglés) entre las tramas g_k y g_{k+1}. Por lo tanto, la maximización de la ecuación (4.16) impone la restricción para que $v(x)$ minimice la DFD. El segundo término en el numerador de la ecuación (4.15) es la pdf condicional del campo de desplazamiento dada la segmentación de movimiento. Este término se modela mediante una distribución de Gibbs de la siguiente manera:

$$p(v_1, v_2|z, g_k) = p(v_1, v_2|z) = \frac{1}{Q_2}\exp\{-U_2(v_1, v_2|z)\} \quad \textbf{(4. 18)}$$

(4.18)

Donde Q_2 es una constante, y $U_2(v_1, v_2|z)$ es el potencial de Gibbs correspondiente, $\|\cdot\|$ denota la distancia euclidiana y N_x es el conjunto de vecindarios del sitio x.

$$\begin{aligned} U_2(v_1, v_2|z) &= \alpha \sum_x \|v(x) - \tilde{v}(x)\|^2 \\ &+ \beta \sum_{x_i} \sum_{x_j \in Nx_i} \|v(x_i) - v(x_j)\|^2 \delta(z(x_i) - z(x_j)) \end{aligned} \quad \textbf{(4. 19)}$$

El primer término de la ecuación (4.19) hace cumplir un cálculo mínimo del vector de movimiento residual $v_r(x)$, esto apunta para minimizar la desviación del campo de movimiento $v(x)$ del campo de movimiento paramétrico $\tilde{v}(x)$ minimizando la DFD. Notar que el campo de movimiento paramétrico $\tilde{v}(x)$ se calcula del conjunto de los parámetros del modelo $a_i, i = 1,...,K$, la cual es una función de $v(x)$ y $z(x)$. El segundo término de la ecuación (4.19) impone una restricción local de alisado *(smoothness piecewise)* en las estimaciones del flujo óptico sin incluir ninguna variable extra como campos de línea. Observar que este término es inactivo sólo para aquellos píxeles en el vecindario N_x el cual comparte la misma etiqueta de segmentación con el sitio x. Por lo tanto el

alisado (*smoothness*) espacial se ve reforzado solo en los vectores de flujo generados por un solo objeto.

El tercer término en la ecuación (4.15) modela la probabilidad *a priori* del campo de segmentación dado por:

$$p(z|g_z) = p(z) = \frac{1}{Q_3}\sum_{\omega\in\Omega}\exp\{-U_3(z)\}\delta(z-\omega) \qquad \textbf{(4. 20)}$$

donde Ω denota el espacio de muestras del vector aleatorio con valor discreto z, Q_3 está dado por la ecuación (4.10) y $U_3(z)$ está dado por:

$$U_3(z) = \sum_{x_i}\sum_{x_j\in Nx_i} V_C\left(z(x_i), z(x_j)\right) \qquad \textbf{(4. 21)}$$

N_x denota el sistema de vecindario para el campo de etiquetas, y

$$V_C\left(z(x_i), z(x_j)\right) = \begin{cases} -\gamma & si \quad z(x_i) = z(x_j) \\ \\ +\gamma & otro \quad caso \end{cases} \qquad \textbf{(4. 22)}$$

La dependencia de las etiquetas en la intensidad de la imagen generalmente no se considera aunque las fronteras de la región generalmente coinciden con los bordes de intensidad.

4.3.3 Algoritmo General

Maximizar la pdf *a posteriori* de la ecuación (4.15) es equivalente a minimizar la función de costo:

$$E = U_1(g_{k+1}|g_k, v_1, v_2, z) + U_2(v_1, v_2|z) + U_3(z) \qquad \textbf{(4. 23)}$$

Esta ecuación esta compuesta por las funciones potenciales dadas por las ecuaciones (4.16), (4.18) y (4.20). Una minimización directa de la ecuación (4.23) respecto a todas las variables no conocidas es un problema difícil de resolver ya que los campos de segmentación y de movimiento contienen un gran conjunto de datos no conocidos.

La minimización de la ecuación (4.23) se puede realizar mediante un proceso iterativo que consiste en dos pasos y el cual fue propuesto por M.M. Chang y M. I. Sezan [28]:

1. Dadas las mejores estimaciones disponibles de los parámetros $a_i, i = 1,...,K$ y z, actualizar el campo de flujo óptico v_1, v_2. Este paso involucra minimizar la siguiente función modificada de costo:

$$E_1 = \sum_x \left[g_k(x) - g_{k+1}\left(x + v(x)\Delta t\right) \right]^2 + \alpha \sum_x \left\| v(x) - \tilde{v}(x) \right\|^2 + \beta \sum_{x_i} \sum_{x_j \in Nx_i} \left\| v(x_i) - v(x_j) \right\|^2 \delta\left(z(x_i) - z(x_j) \right) \quad \textbf{(4. 24)}$$

La ecuación anterior se compone de todos los términos de la ecuación (4.23). El primer término indica qué tan bien $v(x)$ explica nuestras observaciones, el segundo y tercer término imponen restricciones previas en las estimaciones de movimiento que deben estar de acuerdo al modelo de flujo paramétrico y que deben variar suavemente dentro de cada región. Para minimizar esta función de energía se emplea el método HCF (*Highest Confidence First*) propuesto por Chou y Brown [38]. HCF es un método determinístico diseñado para manejar eficientemente la optimización de problemas de múltiples variables con interacciones de vecindario.

2. Actualizar el campo de segmentación z asumiendo que se conoce el campo de flujo óptico $v(x)$. Este paso involucra la minimización de todos los términos en la ecuación (4.23) la cual contiene tanto a z como a $\tilde{v}(x)$ y esta dada por:

$$E_2 = \alpha \sum_x \left\| v(x) - \tilde{v}(x) \right\|^2 + \sum_{x_i} \sum_{x_j \in Nx_i} V_C\left(z(x_i), z(x_j) \right) \quad \textbf{(4. 25)}$$

El primer término en la ecuación (4.25) cuantifica la consistencia de $\tilde{v}(x)$ y $v(x)$. El segundo término está relacionado con la probabilidad *a priori* de la configuración de las etiquetas de segmentación. Se emplea un procedimiento ICM (*Iterated Condicional Modes*) para optimizar E_2 como el propuesto por M.M. Chang, A. M. Tekalp y M. I. Sezan [27].

Se puede encontrar un cálculo inicial del campo de flujo óptico mediante un diseño bayesiano con una restricción de suavizado global *(global smoothness).* Dado este cálculo se pueden inicializar las etiquetas de segmentación con un procedimiento similar al propuesto por Wang y Adelson.

4.4. Formulación de un Modelo para la Segmentación Espacio-temporal

En este capítulo se presenta una estructura probabilística para una segmentación espacio-temporal de secuencias de video. En este diseño desarrollado por Yang Wang et al. [51] la información de movimiento, la información de frontera de la segmentación de intensidad y la conectividad espacial de la segmentación se unifican en el proceso de la segmentación de video por medio de modelos gráficos. Se hace uso de una red bayesiana para modelar las interacciones del campo de vectores de movimiento, el campo de la segmentación de intensidad y el campo de la segmentación de video.
En este modelo propuesto se aplica el concepto de Campos Aleatorios de Markov (CAM) para la formación de regiones continuas. El diseño de la segmentación de video propuesto en este capítulo puede ser visto desde la perspectiva del compromiso de diseños basados en movimiento previo y diseños de fusión de regiones.

4.4.1. Descripción del modelo

La información del movimiento es un elemento fundamental usado para la segmentación de secuencias de video. La escena se puede segmentar en un conjunto de regiones de tal forma que el movimiento de un píxel en cada región sea consistente con un modelo de movimiento o una transformación paramétrica. Algunos ejemplos de modelos de movimiento son el modelo traslacional, el modelo afín y el modelo de perspectiva los cuales utilizan dos, seis y ocho parámetros respectivamente.

Adicional a esto se imponen restricciones espaciales en la región segmentada donde se asume que el movimiento es suave o sigue una transformación paramétrica. En el trabajo desarrollado por M. Chang, I. Sezan y A. Tekalp [29] la información de movimiento y la segmentación son simultáneamente estimadas. La segmentación de intensidad proporciona importantes indicaciones de las fronteras del objeto. Los métodos que combinan la segmentación de intensidad con la información de movimiento se han propuesto en el trabajo de F. Moscheni, S. Bhattacharjee [14] así como en el trabajo propuesto por I. Patras y E. A. Hendriks [20].

Los modelos gráficos proporcionan una herramienta natural para manejar la incertidumbre y la complejidad por medio de un formalismo general para una representación compacta de la distribución de probabilidad conjunta. En particular las redes bayesianas y los Campos Aleatorios de Markov son de gran utilidad en el procesamiento de imágenes y video.
En éste capítulo se presenta un esquema probabilístico en el cual la información espacial y la información temporal interactúan durante el proceso de segmentación de video. Se hace uso de una red bayesiana para modelar las interacciones entre el campo de vectores de movimiento, el campo de segmentación de intensidad y el campo de la segmentación de video. Se aplica

el concepto de Campo Aleatorio de Markov (CAM) para propiciar la conectividad espacial de las regiones segmentadas. Se utiliza un diseño de tres tramas (*three-frame*) para tratar con el problema de las oclusiones.

El criterio de la segmentación es estimar el Máximo a Posteriori (MAP) de los tres campos dadas las tramas consecutivas de video.
Para lograr la optimización es necesario un procedimiento que minimice las funciones objetivas correspondientes de una forma iterativa.

El diseño presentado en este capítulo tiene gran relación con el trabajo de M. Chang *et al.* [27] y el de I. Patras *et al.* [20]. En ambos diseños se estima simultáneamente el campo de vectores de movimiento y el campo de segmentación de video usando un algoritmo MAP-MRF (*Maximum a Posteriori – Markov Random Fields*, por sus siglas en inglés) o bien MAP-CAM (Máximo a Posteriori – Campos Aleatorios de Markov). El método propuesto por Chang adopta un diseño de dos tramas y no hace uso de la restricción impuesta por el campo de segmentación de intensidad durante el proceso de la segmentación de video. Aunque el algoritmo identifica exitosamente múltiples objetos en movimiento dentro de la escena, las fronteras de los objetos son inexactas en sus resultados experimentales.

El método propuesto por I. Patras [20] emplea una segmentación inicial de intensidad y adopta un diseño de tres tramas para tratar con el problema de las oclusiones. Este método conserva las desventajas de los diseños de fusión de regiones. La información temporal puede no actuar sobre la información espacial y la información de frontera no considerada en el campo de la segmentación inicial de intensidad no se puede recuperar por el campo de vectores de movimiento.

Para tratar con las limitaciones de los algoritmos anteriores, el diseño de Yang Wang *et al.* [51] presentado en este capítulo estima simultáneamente los tres campos para obtener resultados espacio-temporalmente coherentes. Las interrelaciones entre los tres campos y los tramas de video sucesivas se describen por un modelo de red bayesiana en donde la información temporal y espacial interactúan una con otra. De ahí que la información de frontera perdida en el campo de segmentación de intensidad pueda ser recuperada por el campo de vectores de movimiento.

4.4.2. Metodología

Representación del modelo

Para una secuencia de imágenes se asume que la intensidad de un píxel permanece constante a lo largo de su trayectoria de movimiento. Las variaciones de iluminación así como las oclusiones del objeto se desprecian, y por lo tanto se puede establecer que:

$$y_k(x) = y_{k-1}(x - d_k(x)) \qquad \textbf{(4. 26)}$$

Donde $y_k(x)$ es la intensidad del píxel dentro de la trama k-ésima del video en el sitio x, con $k \in N$, $x \in X$, y X es el dominio espacial de cada trama de video, $d_k(x)$ es el vector de movimiento desde la trama $k-1$ a la trama k. El campo de vectores de movimiento se expresa de una forma compacta como d_k. Ya que los datos del video son corrompidos durante el proceso de adquisición de la imagen, se requiere un modelo de observación para la secuencia. Asumiendo que el ruido Gaussiano que es independiente e idénticamente distribuido (i.i.d.) corrompe cada punto, entonces el modelo de observación del frame (trama) k-ésimo se representa por:

$$g_k(x) = y_k(x) + n_k(x) \qquad \textbf{(4. 27)}$$

Donde $g_k(x)$ es la intensidad de la imagen observada en el sitio x, y $n_k(x)$ es el ruido aditivo independiente con media cero y varianza σ_n^2. La segmentación de video se refiere a agrupar los píxeles que pertenecen a objetos en movimiento independientes en la escena. Para tratar con las oclusiones se asume que cada sitio x en la trama actual g_k no puede ser ocluido en ambas tramas, es decir, en la trama previa g_{k-1} y en la trama siguiente g_{k+1}. Por lo tanto se adopta y diseño de tres tramas para la segmentación. Teniendo las tres tramas consecutivas de la secuencia de video observada g_{k-1}, g_k, g_{k+1} queremos estimar la distribución de probabilidad condicional conjunta del campo de vectores de movimiento d_k, del campo de segmentación de intensidad s_k y del campo de segmentación de video z_k.

Aplicando la regla de Bayes llegamos a que:

$$p(d_k, s_k, z_k / g_k, g_{k-1}, g_{k+1}) = \frac{p(d_k, s_k, z_k, g_k, g_{k-1}, g_{k+1})}{p(g_k, g_{k-1}, g_{k+1})} \qquad \textbf{(4. 28)}$$

Donde $p(d_k, s_k, z_k / g_k, g_{k-1}, g_{k+1})$ es la función de densidad de probabilidad condicional a posteriori (pdf, por sus siglas en inglés) de los tres campos y el denominador del lado derecho es constante. Las interrelaciones entre $d_k, s_k, z_k, g_k, g_{k-1}, g_{k+1}$ son modeladas en los siguientes aspectos:

1. La estimación de movimiento establece la correspondencia de píxel entre las tres tramas consecutivas. Dada la trama actual y el campo de vectores de movimiento, los píxeles en la trama anterior y en la trama siguiente deben seguir la consideración de intensidad constante en la ecuación (4.26).
2. El campo se segmentación de intensidad proporciona un conjunto de regiones con una variación de intensidad relativamente pequeña en la trama actual.
3. Para identificar objetos independientes en movimiento dentro de la escena, estas regiones se agrupan en segmentos con movimiento coherente
4. Si coexisten múltiples modelos de movimiento dentro de una región, la región debe dividirse en varios segmentos.

Las cuatro interrelaciones anteriormente mencionadas se modelan por las redes bayesianas de la figura 4.1(a)-(d). Combinando las cuatro relaciones, el modelo de segmentación de video se pude representar por la red bayesiana de la figura 4.1(e).

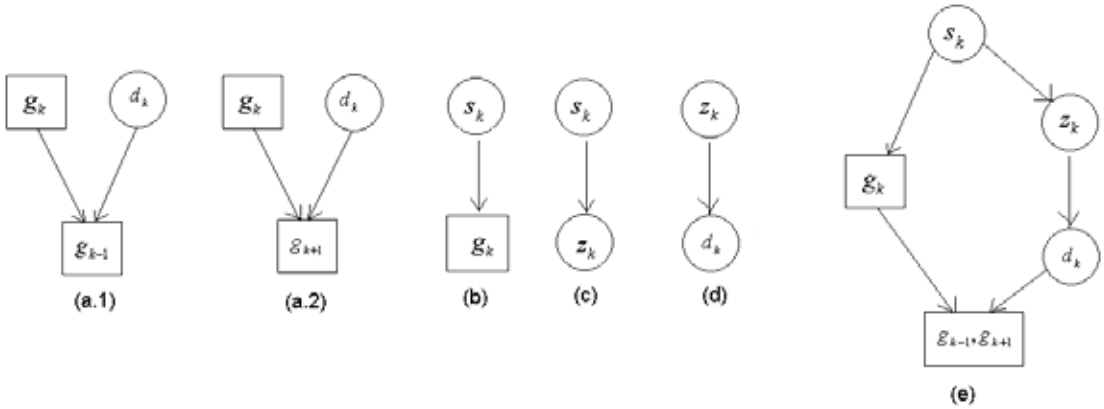

Figura 4.1. Modelo de la red bayesiana para la segmentación de video

Por lo tanto, de acuerdo al campo de vectores de movimiento las regiones en el campo de segmentación de intensidad pueden fusionarse o dividirse para formar segmentos coherentes espacio-temporalmente. Además debe existir conectividad espacial en el proceso de segmentación de video. Las relaciones de independencia condicional implicadas por la red bayesiana permiten representar de una forma compacta la distribución conjunta. Aplicando la regla de la cadena, la densidad de probabilidad conjunta se pude factorizar como el producto de la distribución condicional de cada elemento en la red bayesiana dados sus padres.

$$p(d_k, s_k, z_k, g_k, g_{k-1}, g_{k+1}) = p(g_{k-1}, g_{k+1} / g_k, d_k) \times p(g_k / s_k)\, p(s_k)\, p(d_k / z_k)\, p(z_k / s_k) \quad \textbf{(4. 29)}$$

De ahí que el MAP (Máximo a Posteriori) de los tres campos se convierte en:

$$\begin{aligned}(\hat{d}_k, \hat{s}_k, \hat{z}_k) &= \arg\max_{(d_k, s_k, z_k)} p(d_k, s_k, z_k / g_k, g_{k-1}, g_{k+1}) \\ &= \arg\max_{(d_k, s_k, z_k)} p(d_k, s_k, z_k, g_k, g_{k-1}, g_{k+1}) \\ &= \arg\max_{(d_k, s_k, z_k)} p(g_{k-1}, g_{k+1} / g_k, d_k) \times p(g_k / s_k)\, p(s_k)\, p(d_k / z_k)\, p(z_k / s_k)\end{aligned} \quad \textbf{(4. 30)}$$

Restricciones espacio-temporales

La densidad de probabilidad condicional $p(g_{k-1}, g_{k+1} / g_k, d_k)$ muestra qué tan bien la estimación de movimiento aplica en las tramas consecutivas. Asumiendo que la probabilidad queda completamente especificada por el campo aleatorio de la diferencia de trama (*frame*) desplazada (DFD de sus siglas en inglés), el modelo de observación de video se puede usar para calcular $p(g_{k-1}, g_{k+1} / g_k, d_k)$. Se puede definir la DFD hacia atrás $e_k^b(x)$ y la DFD hacia delante $e_k^f(x)$ en el sitio x como:

$$e_k^b(x) = g_k(x) - g_{k-1}(x - d_k(x)) = n_k(x) - n_{k-1}(x - d_k(x)) \quad \textbf{(4. 31)}$$

$$e_k^f(x) = g_k(x) - g_{k+1}(x + d_k(x)) = n_k(x) - n_{k+1}(x + d_k(x)) \quad \textbf{(4. 32)}$$

El vector $\left(e_k^b(x), e_k^f(x)\right)^T$ se denota como $e_k(x)$. Suponiendo ruido gaussiano i.i.d. sabemos que $e_k(x)$ tiene distribución normal bi-variable con media cero. El coeficiente de correlación de $e_k^b(x)$ y $e_k^f(x)$ es:

$$\rho = \frac{Cov\left[e_k^b(x), e_k^f(x)\right]}{\sqrt{Var\left[e_k^b(x)\right] Var\left[e_k^f(x)\right]}} = \frac{\sigma_n^2}{2\sigma_n^2} = \frac{1}{2} \quad \textbf{(4. 33)}$$

Considerando independencia condicional entre observaciones espacialmente distintas, la densidad de probabilidad se puede factorizar como:

$$\begin{aligned}p(g_{k-1}, g_{k+1} / g_k, d_k) &\approx \prod_{x \in X} p\left(g_{k-1}(x - d_k(x)), g_{k+1}(x + d_k(x)) / g_k(x)\right) \approx \prod_{x \in X} p\left(e_k^b(x), e_k^f(x)\right) \\ &= \left(\frac{1}{2\pi\sqrt{\left|\sum_e\right|}}\right)^{|X|} \exp\left[-\sum_{x \in X} \frac{1}{2} e_k^T(x) \sum_e^{-1} e_k(x)\right] \alpha \exp\left[-\frac{1}{3\sigma_n^2} \sum_{x \in X} U_x^{g|d}\left(d_k(x)\right)\right]\end{aligned} \quad \textbf{(4. 34)}$$

$$U_x^{g|d}\left(d_k(x)\right)=\left(e_k^b(x)\right)^2-2\rho e_k^b(x)e_k^f(x)+\left(e_k^f(x)\right)^2 \quad \textbf{(4. 35)}$$

Donde $\sum_e$ es la matriz de covarianza para cada sitio x, y el coeficiente de correlación ρ se calculó con la ecuación (4.33).

El término $p(g_k/s_k)$ muestra que tan adecuada es la segmentación de intensidad para la escena. Considerando una distribución Gaussiana para cada región segmentada en la trama actual, la densidad de probabilidad condicional se puede factorizar como:

$$p(g_k/s_k)=\prod_{x\in X}p\left(g_k(x)|s_k(x)\right)=\left(\frac{1}{\sqrt{2\pi}\sigma_\eta}\right)^{|X|}\exp\left[-\sum_{x\in X}\frac{1}{2\sigma_\eta^2}\left(g_k(x)-\mu_{s_k(x)}\right)^2\right]$$
$$\alpha\exp\left[-\frac{1}{2\sigma_\eta^2}\sum_{x\in X}U_x^{g|s}(s_k(x))\right] \quad \textbf{(4. 36)}$$

$$U_x^{g|s}\left(s_k(x)\right)=\left(g_k(x)-\mu_{s_k(x)}\right)^2 \quad \textbf{(4. 37)}$$

Donde $s_k(x)=l$ le asigna el sitio x a la región l, μ_l es la intensidad media de la región l, y σ_η^2 es la varianza para cada región.

La función densidad de probabilidad (pdf de $p(s_k)$ representa la probabilidad a priori de la segmentación de intensidad. Para la formación de regiones continuas se modela la densidad $p(s_k)$ con Campos Aleatorios de Markov. Esto es, si N_x es el vecindario de un píxel en x, entonces la distribución condicional de una sola variable en el lugar x depende solo de las variables dentro del vecindario N_x. De acuerdo al teorema Hammersley – Clifford, la densidad está dada por una distribución de Gibbs con la siguiente forma:

$$P(s_k)\alpha\exp\left[-\sum_{c\in C}V_c^S\left(s_k(x)|x\in C\right)\right] \quad \textbf{(4. 38)}$$

Donde C es el grupo de todos los conjuntos c y V_C^S es la función potencial del conjunto. Un conjunto es un grupo de píxeles que son vecinos uno del otro y la función potencial V_C^S depende sólo de los puntos dentro del conjunto C.

Se puede imponer una restricción espacial con el siguiente potencial de conjunto de dos píxeles.

$$V_c^s\left(s_k(x),s_k(y)\right)\alpha U_{x,y}^s\left(s_k(x),s_k(y)\right)=\frac{1}{\|x-y\|^2}\left[1-\delta\left(s_k(x)-s_k(y)\right)\right] \quad \textbf{(4. 39)}$$

Donde:

$$\delta(x)=\begin{cases}1 & si, x=o\\ 0 & otro\end{cases}$$

es la función delta de Kronecker y $\|\cdot\|$ denota la distancia Euclidiana. Por lo tanto es más probable que dos píxeles vecinos pertenezcan a la misma clase que a clases diferentes. La restricción se hace más estricta cuando se disminuye la distancia entre lugares vecinos.

El término $p(d_k / z_k)$ es la densidad de probabilidad condicional del campo de vectores de movimiento dado el campo de segmentación de video. Para aumentar la conectividad espacial, se modela por una distribución de Gibbs con la siguiente función potencial:

$$V_c^{d|z}\left(d_k(x),d_k(y)|z_k\right)\alpha U_{x,y}^{d|z}\left(d_k(x),d_k(y),z_k(x),z_k(y)\right)$$
$$=\frac{1}{\|x-y\|^2}\delta\left(z_k(x)-z_k(y)\right)\|d_k(x)-d_k(y)\|^2 \qquad \textbf{(4. 40)}$$

La restricción de suavizado (*conocida como pairwise smoothness*) de los vectores de movimiento se impone sólo cuando los dos puntos de vecindad comparten la misma etiqueta de segmentación de video. Esto ocasiona que una región se divida en varios segmentos cuando diferentes modelos de movimiento coexisten. Por eso, $U_{x,y}^{d|z}$ se puede ver como la fuerza de división de región.

El último término $p(z_k / s_k)$ representa la densidad de probabilidad *a posteriori* del campo de la segmentación de video cuando se tiene el campo de la segmentación de intensidad.

La densidad se modela por una distribución de Gibbs con la siguiente función potencial:

$$V_c^{z|s}\left(z_k(x),z_k(y)|s_k\right)\alpha U_{x,y}^{z|s}\left(z_k(x),z_k(y),s_k(x),s_k(y)\right)$$
$$=\frac{1}{\|x-y\|^2}\left[1-\delta\left(z_k(x)-z_k(y)\right)\right]+\frac{\alpha}{\|x-y\|^2}\delta\left(s_k(x)-s_k(y)\right)\left[1-\delta\left(z_k(x)-z_k(y)\right)\right] \qquad \textbf{(4. 41)}$$

El primer término del lado derecho propicia la conectividad espacial de la segmentación de video mientras que el segundo término propicia que píxeles vecinos compartan la misma etiqueta de segmentación de video cuando se encuentran dentro de una región del campo de segmentación de intensidad. Por lo tanto, $U_{x,y}^{z|s}$ propicia que las regiones de la segmentación de intensidad se agrupen. El parámetro α controla la restricción impuesta por la segmentación de intensidad. Las interacciones en la red Bayesiana se modelan por las restricciones espacio-temporales anteriormente mencionadas. Combinando estos términos pdf (función densidad de probabilidad) el criterio de la estimación MAP se convierte en:

$$(\hat{d}_k, \hat{s}_k, \hat{z}_k) = \arg \min_{(d_k, s_k, z_k)} \left[\begin{array}{l} \sum_{x \in X} U_x^{g|d}\left(d_k(x)\right) + \lambda_1 \sum_{x \in X} U_x^{g|s}\left(s_k(x)\right) + \lambda_2 \sum_{\{x,y\} \in C} U_{x,y}^{s}\left(s_k(x), s_k(y)\right) \\ + \lambda_3 \sum_{\{x,y\} \in C} U_{x,y}^{d|z}\left(d_k(x), d_k(y), z_k(x), z_k(y)\right) \\ + \lambda_4 \sum_{\{x,y\} \in C} U_{x,y}^{z|s}\left(z_k(x), z_k(y), s_k(x), s_k(y)\right) \end{array} \right]$$

(4. 42)

Donde los parámetros $\lambda_1, \lambda_2, \lambda_3$ y λ_4 controlan la contribución de los términos individuales.

En este modelo, la segmentación de video está influenciada por la información espacial y la información temporal. Se debe hacer notar que la dirección de las flechas en el modelo de red bayesiana no significa que la influencia entre causa y consecuencia sea en un solo sentido.

La trama de video actual puede ser pensada como la causa de la siguiente trama. Para una secuencia de imágenes, tanto la secuencia original como la secuencia en orden inverso son comprendidas desde el punto de vista de segmentación (en la secuencia inversa, la aparición de objetos y las relaciones de oclusión son las mismas que en la secuencia original, mientras que los modelos de movimiento se invierten para todos los objetos en la escena). Por lo tanto la trama actual también puede ser vista como la causa de la trama previa (en la secuencia invertida). En este modelo [51], g_k es la causa tanto de la trama siguiente g_{k+1} como de la trama previa g_{k-1}.

El campo de vectores de movimiento establece la correspondencia entre la trama actual y sus dos tramas vecinas. Cuando la trama g_{k+1} y la trama g_{k-1} se separan, como se observa en la figura 4.2, las interrelaciones se ven más claras. Por lo tanto, de la estructura de la red bayesiana, sabemos que;

$$p(g_{k-1}, g_{k+1} | g_k, d_k) = p(g_{k-1} | g_k, d_k)\, p(g_{k+1} | g_k, d_k) = \prod_{x \in X} p\left(e_k^b(x)\right) p\left(e_k^f(x)\right)$$

$$\alpha \exp\left[-\frac{1}{4\sigma_n^2} \sum_{x \in X} \left(e_k^b(x)\right)^2 + \left(e_k^f(x)\right)^2 \right] \qquad \textbf{(4. 43)}$$

Comparando con la ecuación (4.34), el coeficiente de correlación de $e_k^b(x)$ y $e_k^f(x)$ es cero en (4.43). La red bayesiana en la figura 4.2 no considera la interacción entre la DFD hacia delante y la DFD hacia atrás. Por lo tanto, el modelo de red bayesiana de la figura 4.2 es solo una simplificación del modelo original. En la ecuación (4.40) cuando el parámetro α vale cero, la restricción de la segmentación de intensidad desaparece por lo que el método se degenera en

un diseño basado en movimiento. Cuando α tiende a infinito, las fronteras en el campo de segmentación de video deben provenir del campo de segmentación de intensidad y la técnica se convierte en un diseño de fusión de regiones. Por lo tanto, el método propuesto se puede ver como un compromiso de diseños basados en el movimiento previo y diseños basados en fusión de regiones.

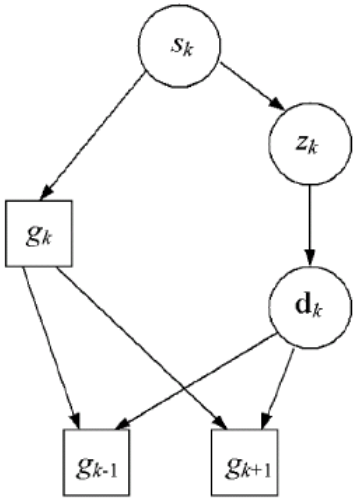

Figura 4.2. Modelo simplificado de la red bayesiana para la segmentación de video

4.4.3. Esquema de Optimización

No existe un método simple para minimizar directamente la ecuación (4.42). Para eso, se propone una estrategia de optimización iterativa sobre los siguientes dos pasos:

Primero, se actualiza d_k y s_k dado el cálculo del campo de segmentación de video z_k. De la estructura de la red bayesiana propuesta se puede ver que d_k y s_k son condicionalmente independientes cuando el campo de segmentación de video z_k y las tres tramas sucesivas están dados.

La estimación conjunta se puede factorizar como:

$$(\hat{d}_k, \hat{s}_k) = \arg\max_{(d_k, s_k)} p(d_k, s_k | g_k, g_{k-1}, g_{k+1}, \hat{z}_k) =$$

$$\left(\arg\max_{d_k} p(d_k | g_k, g_{k-1}, g_{k+1}, \hat{z}_k) \arg\max_{s_k} p(s_k | g_k, \hat{z}_k) \right) \qquad \textbf{(4. 44)}$$

Usando la regla de la cadena, la estimación MAP se convierte en:

$$\hat{d}_k = \arg\max_{d_k} p(d_k | g_k, g_{k-1}, g_{k+1}, \hat{z}_k) = \arg\max_{d_k} p(g_{k-1}, g_{k+1} | g_k, d_k)\, p(d_k | \hat{z}_k)$$

(4. 45)

$$\hat{s}_k = \arg\max_{s_k} p(s_k | g_k, \hat{z}_k) = \arg\max_{s_k} p(g_k | s_k)\, p(s_k)\, p(\hat{z}_k | s_k) \qquad \textbf{(4. 46)}$$

Después, se actualiza z_k dada la estimación del campo de movimiento d_k y el campo de segmentación de intensidad s_k.

$$\hat{z}_k = \arg\max_{z_k} p(z_k | g_k, g_{k-1}, g_{k+1}, \hat{d}_k, \hat{s}_k) = \arg\max_{z_k} p(z_k | \hat{d}_k, \hat{s}_k)$$
$$= \arg\max_{z_k} p(\hat{d}_k | z_k)\, p(z_k | \hat{s}_k) \qquad \textbf{(4. 47)}$$

En este diseño, se usa el sistema de vecindad de 24 puntos (Sistema vecino de quinto orden) y los potenciales se definen solo en *cliques* de dos puntos. Usando los términos de la ecuación (4.42), la estimación MAP bayesiana en las ecuaciones (4.45), (4.46) y (4.47) se puede obtener minimizando las siguientes funciones objetivas:

$$F^d(d_k) = \sum_{x \in X} \left[U_x^{g|d}(d_k(x)) + \frac{1}{2}\lambda_3 \sum_{y \in Nx} U_{x,y}^{d|z}(d_k(x), d_k(y), \hat{z}_k(x), \hat{z}_k(y)) \right] \qquad \textbf{(4. 48)}$$

$$F^s(s_k) = \sum_{x \in X} \left[\lambda_1 U_x^{g|s}(s_k(x)) + \frac{1}{2}\lambda_2 \sum_{y \in Nx} U_{x,y}^{s}(s_k(x), s_k(y)) + \right.$$
$$\left. \frac{1}{2}\lambda_4 \sum_{y \in Nx} U_{x,y}^{z|s}(\hat{z}_k(x), \hat{z}_k(y), s_k(x), s_k(y)) \right] \qquad \textbf{(4. 49)}$$

$$F^z(z_k) = \sum_{x \in X} \left[\frac{1}{2}\lambda_3 \sum_{y \in Nx} U_{x,y}^{d|z}(\hat{d}_k(x), \hat{d}_k(y), z_k(x), z_k(y)) \right.$$
$$\left. + \frac{1}{2}\lambda_4 \sum_{y \in Nx} U_{x,y}^{z|s}(z_k(x), z_k(y), \hat{s}_k(x), \hat{s}_k(y)) \right] \qquad \textbf{(4. 50)}$$

Donde N_x es la vecindad del píxel en x.

Optimización Local

Por lo general, las funciones objetivas no son convexas y no tienen un mínimo único. Se usa el algoritmo de modos condicionales iterados (ICM – *Iterated Conditional Modes*) para llegar a una estimación sub-óptima de cada función objetiva. El algoritmo ICM emplea una estrategia de minimización iterativa. Dados los datos observados y otras etiquetas calculadas, la etiqueta de

segmentación es secuencialmente actualizada minimizando localmente la función objetiva en cada sitio.

Para emplear efectivamente los indicios de frontera de la información espacial en la optimización local se realiza una transformación de distancia en el campo de segmentación de intensidad. Cada píxel x en la imagen transformada en distancia tiene un valor $d_x(s_k)$ que representa la distancia entre el píxel y el píxel de frontera más cercano en s_k. Aquí, un píxel "x" de frontera tiene al menos un punto "y" dentro de su vecindad donde $s_k(y)$ no es lo mismo que $s_k(x)$. El término $U_{x,y}^{z/s}$ en la ecuación (4.49) se remplaza por:

$$U'^{z|s}_{x,y}\left(z_k(x), z_k(y), \hat{s}_k(x), \hat{s}_k(y)\right) = \frac{1}{\|x-y\|^2}\left[1-\delta\left(z_k(x)-z_k(y)\right)\right]$$
$$+\frac{\alpha\theta\left(d_x(\hat{s}_k)-d_y(\hat{s}_k)\right)}{\|x-y\|^2}\times\delta\left(\hat{s}_k(x)-\hat{s}_k(y)\right)\left[1-\delta\left(z_k(x)-z_k(y)\right)\right] \qquad \textbf{(4. 51)}$$

Donde:

$$\theta(x)=\begin{cases}2, & if & x<0\\ 1, & if & x=0\\ 0, & cualquier & otro\end{cases}$$

Si dos píxeles vecinos dentro de una región de segmentación de intensidad no comparten la misma etiqueta, el término θ ayuda a dar una penalidad al campo de segmentación de intensidad en el píxel más cercano a la frontera. Se debe notar que $U'^{z|s}_{x,y}$ no destruye la simetría del potencial del *clique* de dos píxeles en el CAM (Campo Aleatorio de Markov). $U'^{z|s}_{x,y}$ se asocia con la función objetiva de la ecuación (4.50) y con el algoritmo de optimización.

El algoritmo de optimización actualiza la etiqueta minimizando localmente la función objetiva en cada sitio. Se considera un potencial en ambos sitios. Para la función objetiva, $U'^{z|s}_{x,y}$ es equivalente a $U^{z|s}_{x,y}$ ya que la penalidad total para el campo completo es la misma. $U^{z|s}_{x,y}$ es simétrica y cumple con la definición de CAM. La diferencia entre $U'^{z|s}_{x,y}$ y $U'^{z|s}_{x,y}$ ocurre en la minimización local del proceso de optimización.

Se prefiere la forma de la ecuación (4.51), ya que en algunos experimentos realizados, la información de frontera se calcula más exactamente dando la penalidad total al sitio cercano a la frontera en vez de asignar uniformemente la penalidad a ambos lados en la optimización local.

De forma similar, en la ecuación (4.49), $U_{x,y}^{z|s}$ puede ser remplazada por:

$$U_{x,y}^{"z|s}\left(\hat{z}_k(x),\hat{z}_k(y),s_k(x),s_k(y)\right)= \frac{\alpha\theta\left(d_x(\hat{z}_k)-d_y(\hat{z}_k)\right)}{\left\|x-y\right\|^2}\times\delta\left(s_k(x)-s_k(y)\right)\left[1-\delta\left(\hat{z}_k(x)-\hat{z}_k(y)\right)\right] \quad \textbf{(4. 52)}$$

Comparada con la ecuación (4.41), la ecuación (4.52) no considera el primer término de la ecuación (4.41) ya que éste es constante cuando se da el campo de la segmentación de video. Por lo tanto, se obtienen las funciones objetivas locales reales que se optimizan secuencialmente en cada sitio.

$$F_x^d(d_k)=U_x^{g|d}\left(d_k(x)\right)+\frac{1}{2}\lambda_3\sum_{y\in Nx}U_{x,y}^{d|z}\left(d_k(x),d_k(y),\hat{z}_k(x),\hat{z}_k(y)\right) \quad \textbf{(4. 53)}$$

$$F_x^s(s_k)=\lambda_1 U_x^{g|s}\left(s_k(x)\right)+\frac{1}{2}\lambda_2\sum_{y\in Nx}U_{x,y}^{s}\left(s_k(x),s_k(y)\right) +\frac{1}{2}\lambda_4\sum_{y\in Nx}U'^{z|s}_{x,y}\left(\hat{z}_k(x),\hat{z}_k(y),s_k(x),s_k(y)\right) \quad \textbf{(4. 54)}$$

$$F_x^z(z_k)=\frac{1}{2}\lambda_3\sum_{y\in Nx}U_{x,y}^{d|z}\left(\hat{d}_k(x),\hat{d}_k(y),z_k(x),z_k(y)\right) +\frac{1}{2}\lambda_4\sum_{y\in Nx}U'^{z|s}_{x,y}\left(z_k(x),z_k(y),\hat{s}_k(x),\hat{s}_k(y)\right) \quad \textbf{(4. 55)}$$

Inicialización y parámetros

El campo de segmentación de intensidad se inicializa usando un algoritmo de agrupamientos (*clustering) k-means* para incluir la restricción espacial. Cada segmento se caracteriza por una intensidad constante y la restricción espacial se impone por un *clique* potencial de dos puntos, el cual es una simplificación del algoritmo de agrupamiento (*clustering*) adaptivo propuesto por Papps [48]. El campo de vectores de movimiento se inicializa con la estimación MAP con una restricción *pairwise smoothness*.

Wang y Adelson han propuesto un método para la inicialización del campo de segmentación de video cuando se tiene una estimación inicial del movimiento. La trama actual se divide en bloques más pequeños y para el movimiento de cada uno se calcula una transformación afín. Se calcula un conjunto de modelos de movimiento agrupando adaptativamente los parámetros afines. Después se asignan las etiquetas de segmentación de video de forma que minimicen la distorsión del movimiento. En el trabajo desarrollado por Yang Wang et al. [51] el campo de segmentación se inicializa combinando el procedimiento

anteriormente descrito con una restricción espacial *pairwise* en la asignación de regiones.

Para la selección de los parámetros se emplea la idea propuesta por Chang [29]. Después de la inicialización de los tres campos, los parámetros $\lambda_1, \lambda_2, \lambda_3$ y λ_4 se determinan igualando las contribuciones de los potenciales en las funciones objetivas. Primero se calcula λ_3 balanceando los dos potenciales en la ecuación (4.48). Después se pude calcular λ_4 balanceando los dos potenciales en la ecuación (4.50) y finalmente se encuentran λ_1 y λ_2 balanceando los tres potenciales en la ecuación (4.49). El parámetro α en la ecuación (4.51) controla la restricción impuesta por el campo de segmentación de intensidad. Una mejor información de las fronteras del objeto se obtiene con la segmentación de intensidad, se debe tener una mayor penalidad, es decir un valor mayor de α, cuando los bordes del objeto en el campo de segmentación de video no proviene del campo de segmentación de intensidad. Para una secuencia individual, el parámetro α se determina manualmente. En los experimentos realizados por M.M. Chang *et al.* [27], se obtuvieron de forma empírica los valores de α entre 0.5 y 2 para una segmentación de video robusta. Por otra parte, el tamaño del vecindario también influye en la restricción espacio-temporal. Los resultados de la segmentación tendrán mucho ruido o estarán *sobre-suavizados* si el tamaño del vecindario es excesivamente pequeño o excesivamente grande. Con un vecindario de 24 píxeles (5X5) se obtienen mejores resultados para la segmentación de video que con vecindarios de 8 píxeles (3X3) y de 48 píxeles (7X7). Los resultados de la segmentación también dependerán del tamaño de la imagen en cada trama de la secuencia de video y de la escala de los objetos.

4.4.4. Resultados del diseño

En esta sección se presentan los resultados obtenidos por el algoritmo propuesto por Wang *et al.* [51] el cual es una extensión del método de estimación de movimiento y segmentación simultáneos mostrado en el capítulo 4 (sección 4.3.3) agregando otras características que mejoran los resultados obtenidos sobre todo en el campo de segmentación de movimiento.

Se presentarán los resultados obtenidos por Yang Wang et al. [51]. También se comentan los resultados obtenidos por la implementación del algoritmo propuesto en el capítulo 6 donde se realiza una segmentación supervisada de secuencias de video digital a color, los resultados fueron mostrados en dicho capítulo.

El Algoritmo de estimación de movimiento y segmentación simultáneos presentado en el capítulo 4 (sección 4.3.3) no solo permite la estimación de movimiento 3-D en presencia de múltiples objetos en movimiento sino que también proporciona mejores estimaciones del flujo óptico. Varios algoritmos de

análisis del movimiento se pueden formular como casos especiales de éste diseño. Si solo se conservan el primer y tercer término de la ecuación (4.23) y se asume que todos los sitios tienen la misma etiqueta de segmentación entonces se tiene una estimación Bayesiana del movimiento con una restricción de alisado global (*global smoothness*).

El algoritmo de estimación del movimiento y etiquetamiento de región propuesto por C. Stiller [7] involucra todos los términos de la ecuación (4.23) excepto el primero. Las etiquetas de segmentación en el algoritmo de Stiller se usan solamente como señales que permiten una restricción *piecewise smoothness* en el flujo óptico y no intenta hacer cumplir la consistencia de los vectores de flujo con una componente paramétrica.

El algoritmo de estimación del movimiento de Dubois y Honrad [12] el cual utiliza campos de línea es fundamentalmente diferente en que modela las discontinuidades en el campo de movimiento en vez de modelar regiones que correspondan a movimientos físicos diferentes.

Por otro lado el algoritmo de segmentación de movimiento propuesto por Murray y Buxton [10] emplea sólo el segundo término de la ecuación (4.19) y solo el tercer término de la ecuación (4.23) para modelar la pdf condicional y a priori respectivamente. El diseño propuesto por Wang y Adelson sólo usa el primer término de la ecuación (4.29) para calcular la segmentación de movimiento.

Los algoritmos de Wang-Adelson (W-A) [2] y Murray-Buxton (M-B) [2] utilizan el flujo óptico calculado por el algoritmo de Horn-Schunck [2] como entrada. Se debe fijar el número de regiones así como el tamaño de los bloques para encontrar los parámetros del movimiento afín en el algoritmo de Wang-Adelson (W-A) para que después dichos parámetros afines calculados para cada bloque se agrupen usando un algoritmo *k-means*. Es importante que el tamaño de los bloques sea lo suficientemente grande para identificar el movimiento rotacional.

Pueden utilizarse los resultados arrojados por el algoritmo W-A como entrada al algoritmo M-B, éste último elimina las regiones pequeñas aisladas.

Otro diseño es inicializar un algoritmo de estimación de movimiento y segmentación simultáneas MAP con los resultados obtenidos con el algoritmo de Wang-Adelson. Un diseño de este tipo fue presentado en este capítulo. En dicho diseño propuesto por Yang Wang *et al*. [51] se obtienen muy buenos resultados de la segmentación de objetos. Los resultados probados en al secuencia "*Flower Garden"* y en la secuencia "*table tennis"* se muestran en la figura y fueron tomados de dicho artículo. Para éste caso ellos asumen que existen cuatro objetos en el campo de segmentación de video.

El campo de vectores de movimiento, el campo de segmentación de intensidad y el campo de segmentación de video son recuperados utilizando la técnica propuesta y explicada ampliamente a lo largo de este capítulo.

Figura 4.3. Una trama de la secuencia *"flower garden"* tomada de [51]

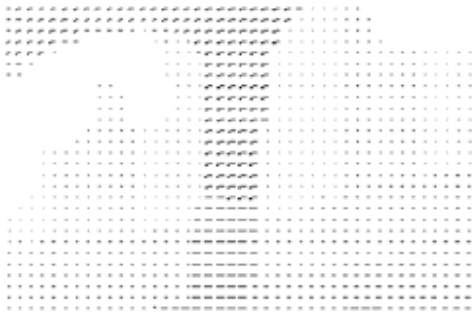

Figura 4.4. Campo de vectores de movimiento. Tomada de [51]

Figura 4.5. Campo de segmentación con 4 niveles de intensidad. Tomada de [51]

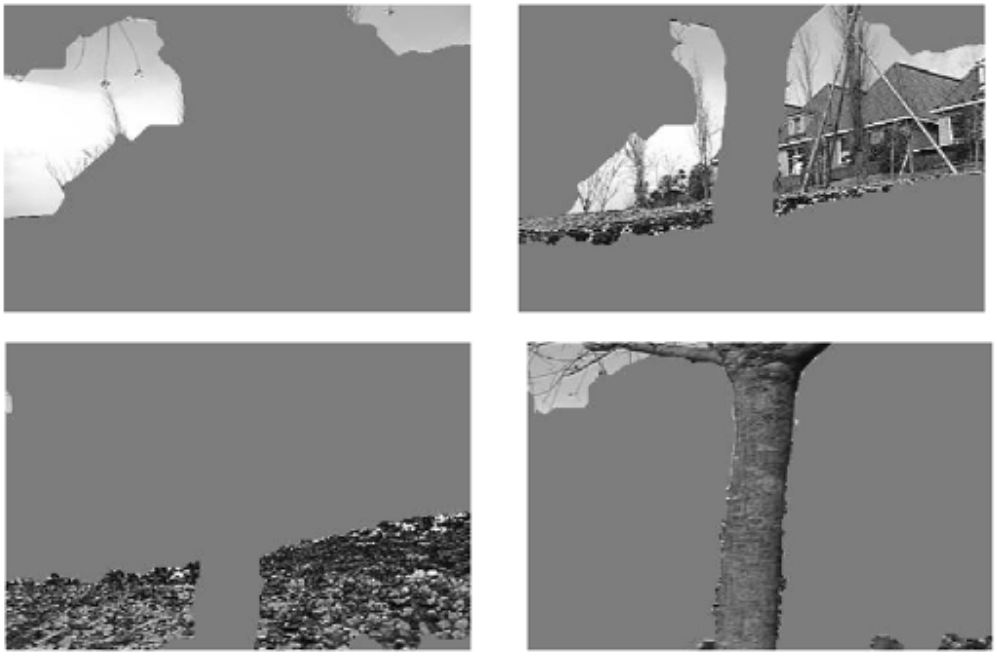

Figura 4.6. Resultados de la segmentación de video. Tomada de [51].

Del campo de vectores de movimiento mostrado en la figura 4.4 se puede observar que el problema de las oclusiones del movimiento se resuelve de una forma satisfactoria debido al uso de tres tramas. En la figura 4.5 se muestra el resultado de una segmentación de 4 niveles de intensidad donde un área con intensidad constante representa un segmento de intensidad. En la secuencia *"flower garden"* se conserva la información de los bordes en el campo de segmentación de intensidad. El algoritmo es capaz de distinguir diferentes objetos en movimiento dentro de una misma escena agrupando exitosamente las regiones pequeñas que son coherentes espacio-temporalmente.

En la secuencia *"table tennis"* de la figura 4.7 la información perdida en el campo de segmentación de intensidad se recupera de acuerdo a la información aportada por el campo de vectores de movimiento, sin embargo las fronteras se detectan con mayor exactitud cuando las características espaciales se relacionan con las características temporales.

Este algoritmo de segmentación es robusto incluso en aquellas áreas grandes homogéneas donde existe poca información de movimiento, como por ejemplo en el cielo de la primera imagen de la figura 4.6 o en la mesa de la secuencia *"table tennis"*

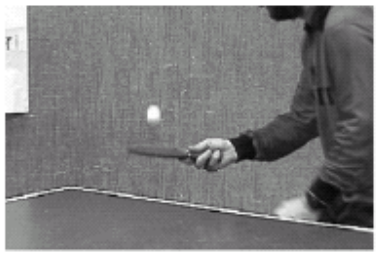

Figura 4.7. Una trama de la secuencia *"table tennis".* Tomada de [51].

Figura 4.8. Campo de vectores de movimiento para la secuencia *"table tennis".* Tomada de [51].

Figura 4.9. Campo de segmentación con 4 niveles de intensidad. Tomada de [51].

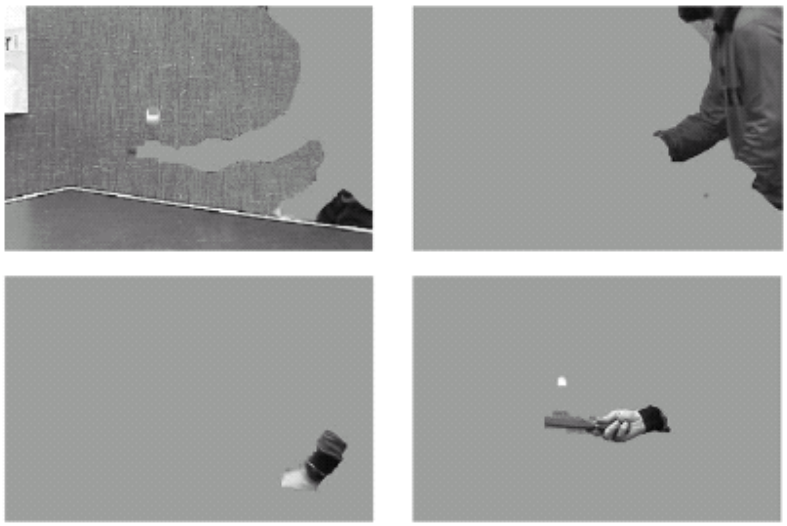

Figura 4.10. Resultados de la segmentación de video. Tomada de [51].

CAPÍTULO 5

5. Diseño, implementación y resultados de un modelo de dos capas (background/foreground) para la segmentación espacio-temporal

Introducción

En este capítulo se presenta un algoritmo capaz de separar una imagen en movimiento del fondo en secuencias de video monoculares. La segmentación automática de capas a partir solamente del color/contraste o solamente del movimiento es propensa de error. Aquí, las señales de movimiento, de color y de contraste interactúan de forma probabilística con información espacial y temporal para segmentar objetos de forma exacta y eficiente.

En este algoritmo no se requieren las velocidades de píxel ya que no se necesita la estimación de flujo óptico debido a que la principal aplicación de este algoritmo es en la segmentación de video en tiempo real como video conferencias lo cual no podría ser posible con el campo de vectores de movimiento debido a la dependencia de la segmentación de una buena estimación del movimiento y debido también al costo computacional que se traduce en tiempo de procesamiento.

En vez de lo anterior, el movimiento eficiente se hace operar directamente y conjuntamente con el cambio de intensidad y el contraste. La salida se hace interactuar después con la información de color. La segmentación se representa mediante un CAM (Campo Aleatorio de Markov) espacial favoreciendo la coherencia excepto donde existe un contraste alto. Finalmente, se logra de forma eficiente una segmentación exacta de capa y una detección de oclusiones mediante un corte binario del gráfico, A. Criminisi **¡Error! No se encuentra el origen de la referencia.**.

En este capítulo se trata la forma de extraer un objeto en primer plano de un fondo estático en un video en tiempo real. Una aplicación de este algoritmo es en la sustitución del fondo en una videoconferencia. El reto de este algoritmo es realizar la separación en capas con una buena calidad de los gráficos de computadora y con una eficiencia aceptable para lograr una velocidad de video en vivo.

Como se presento en los capítulos a lo largo de esta tesis, la segmentación basada en el movimiento se ha logrado mediante la estimación del flujo óptico (velocidades de los píxeles) y después se procede a agrupar los píxeles en regiones de acuerdo a modelos de movimiento predefinidos. Sin embargo, el principio de agrupamiento generalmente requiere algunas consideraciones acerca de la naturaleza del movimiento (traslacional, afín, entre otros) lo cual es restrictivo. Además las soluciones basadas en la estimación de flujo óptico tienden a introducir inexactitudes no deseadas a lo largo de las fronteras de los

objetos. Por último, una estimación exacta del flujo óptico es computacionalmente costosa requiriendo una búsqueda exhaustiva en el vecindario de cada punto (píxel).
En el algoritmo propuesto en éste capítulo, se evita una estimación explícita de las velocidades de píxel. En vez de ella, se emplea un modelo discriminativo eficiente para separar el movimiento usando derivadas espacio-temporales.

En desarrollos recientes, las técnicas de segmentación iterativas que explotan las señales de color/contraste han demostrado ser muy eficientes para imágenes estáticas. En el diseño propuesto en este capítulo, también se agrega la coherencia temporal para incrementar la exactitud de la segmentación y las probabilidades de las transiciones temporales se modelan con una detección explícita de las oclusiones temporales.

Notación e imagen observable

Como se verá mas adelante, en este diseño se modelan las probabilidades del color del fondo y del primer plano de forma no paramétrica usando el espacio de color YUV ya que los modelos de GMM (Gaussian Mixture Models) en el espacio RGB donde las mezclas del fondo y el primer plano se conocen por medio del algoritmo de la Maximización de la Expectativa (EM) presentan algunos problemas en la etapa de inicialización.

Dada una secuencia de imágenes de entrada, una trama se representa como un arreglo $z = (z_1, z_2, ..., z_n, ..., z_N)$ de píxeles en el espacio de color YUV marcado por el subíndice n. La trama en el tiempo *t* se denota con z^t. Las derivadas temporales se denotan con:
Y en cada tiempo *t*, se calculan como:

$$\dot{z} = \left(\dot{z}_1, \dot{z}_2, ..., \dot{z}_n, ..., \dot{z}_N\right) \quad \textbf{(5. 1)}$$

$$\dot{z}_n^t = \left| G\left(z_n^t\right) - G\left(z_n^{t-1}\right) \right| \quad \textbf{(5. 2)}$$

Con $G(\cdot)$ una Gaussiana kernel en la escala de σ_t píxeles. También los gradientes espaciales se calculan convolucionando las imágenes con derivada de primer orden de Gaussiana kernel con desviación estándar σ_s.
Aquí se usa $\sigma_s = \sigma_t = 0.8$, aproximando a un filtro con muestreo de Nyquist. Las derivadas espacio-temporales se calculan en el canal del espacio de color Y. Los movimientos observables se denotan como:

$$m = (g, \dot{z}) \quad \textbf{(5. 3)}$$

Y se utilizan como las características de la imagen para la diferenciación entre el movimiento y el *stasis.*

La segmentación se expresa como un arreglo de valores de opacidad $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n, ..., \alpha_N)$. Nos enfocaremos en la segmentación binaria, $\alpha \in \{F, B\}$, donde F denota el *foreground* y B denota el *background*.

5.1. Modelo probabilístico de segmentación

Este modelo se basa en la minimización de la energía. Se extienden los modelos de energía previos agregando un segundo orden temporal, la cadena previa de Markov y una probabilidad de la observación para el movimiento de la imagen. El modelo posterior es un campo aleatorio condicional (*CRF – Conditional Random Field*, por sus siglas en inglés) con una factorización que contiene alguna estructura generativa reconocible y se utiliza para determinar las formas algebraicas precisas de los factores. Después se fijan varios parámetros de forma discriminada. El campo aleatorio condicional (CRF) se denota como:

$$p\left(\alpha^1, ..., \alpha^t \middle| z^1, ..., z^t, m^1, ..., m^t\right) \propto \exp - \left\{ \sum_{t'=1}^{t} E^{t'} \right\} \qquad \textbf{(5. 4)}$$

Donde

$$E^t = E\left(\alpha^t, \alpha^{t-1}, \alpha^{t-2}, z^t, m^t\right) \qquad \textbf{(5. 5)}$$

El objetivo principal es estimar $\alpha^1, ..., \alpha^t$ dada la imagen y los datos del movimiento, en un principio esto podrá ser realizado mediante la maximización conjunta a posteriori, o equivalentemente a la minimización de la energía:

$$(\hat{\alpha}^1, ..., \hat{\alpha}^t) = \arg\min \sum_{t'=1}^{t} E^{t'} \qquad \textbf{(5. 6)}$$

Sin embargo, tal cálculo no es de interés para las aplicaciones en tiempo real debido a que causan restricción – cada $\hat{\alpha}^{t'}$ debe ser entregada con la evidencia de su pasado sin usar ninguna evidencia del futuro. Por lo tanto la estimación se realizará por una minimización separada de cada término E^t.

5.1.1. Términos de energía del Campo Aleatorio Condicional

La energía E^t asociada con el tiempo *t* es una suma de términos en cuya probabilidad y previos no están completamente separados, y por lo tanto no representan un modelo generativo puro, aunque algunos de los términos tienen interpretaciones claramente generativas. La energía se descompone como la suma de cuatro términos:

$$E\left(\alpha^t, \alpha^{t-1}, \alpha^{t-2}, z^t, m^t\right) = V^T\left(\alpha^t, \alpha^{t-1}, \alpha^{t-2}\right) + V^S\left(\alpha^t, z^t\right) + U^C\left(\alpha^t, z\right) + U^M\left(\alpha^t, \alpha^{t-1}, m^t\right) \quad \textbf{(5. 7)}$$

Donde los dos primeros términos son términos cuya información es tomada del video, es decir, es información previa (*prior-like)* y los segundos dos son probabilidades de observación. El papel de los cuatro términos es el siguiente:

El término previo temporal $V^T(...)$ es una cadena de segundo orden de Markov que impone una tendencia a la continuidad temporal de las etiquetas de segmentación.

El término previo espacial $V^S(...)$ es un término que impone una tendencia a la continuidad espacial de las etiquetas y es inhibido por un alto contraste.

El término de probabilidad del color $U^C(...)$ evalúa la evidencia de las etiquetas de píxel basado en la distribución de colores en el primer plano (*foreground*) y el fondo *(background)*.

El término de probabilidad del movimiento $U^M(...)$ evalúa la evidencia para las etiquetas de píxel basada en la expectativa del estado estático (*stasis*) en el fondo y movimiento que frecuentemente ocurre en el primer plano (*foreground*). Notar que el movimiento m^t se explica en términos del etiquetado tanto de la trama actual α^t como el de la trama previa α^{t-1}.

La figura 5.1 muestra el modelo gráfico espacio-temporal de *Hidden Markov* que representa la probabilidad del color así como la probabilidad del movimiento junto con la información previa temporal y espacial. La misma cadena temporal se repite en cada posición del píxel.

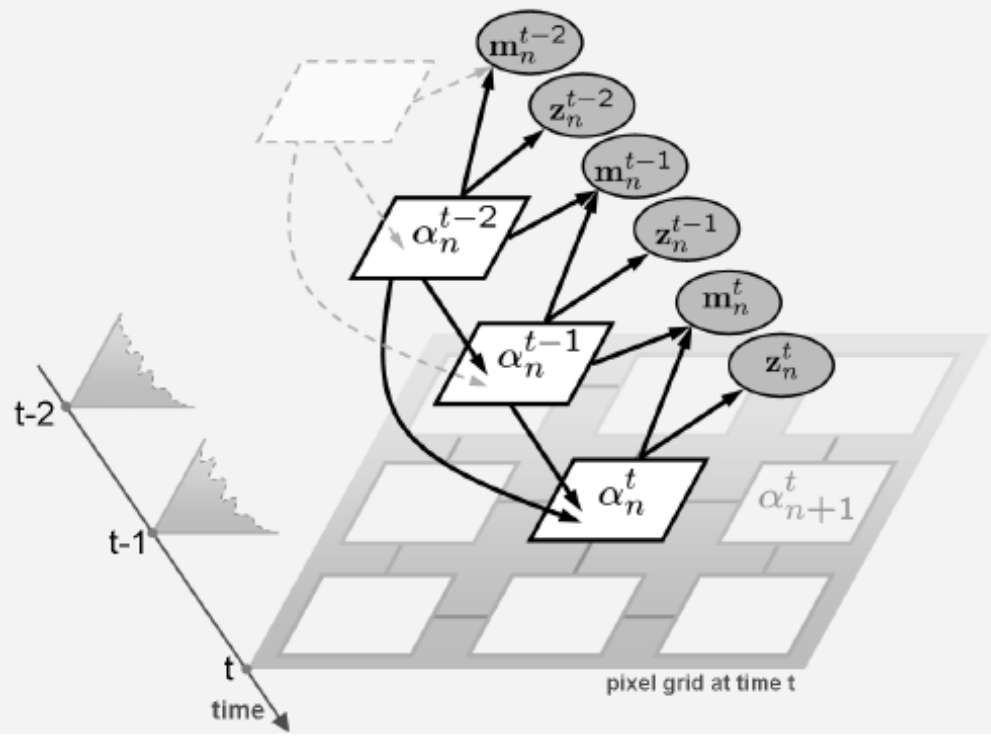

Figura 5.1. Modelo espacio-temporal de Hidden Markov [1].

5.1.2. Término temporal previo

La figura 5.2 Ilustra los cuatro diferentes tipos de transiciones temporales por los que un píxel puede pasar en una escena de dos capas con base en un análisis de dos-tramas. Por ejemplo, un píxel del primer plano puede permanecer en dicho primer plano (píxel etiquetado con FF en la figura 5.2 ó puede moverse al fondo (píxel etiquetado con FB), y así sucesivamente. El punto crítico aquí es que una cadena de Markov de primer orden no es adecuada para transportar la naturaleza de la coherencia temporal en este problema, por lo que se requiere una cadena de Markov de segundo orden. Por ejemplo, un píxel que estaba en el fondo en el tiempo $t-2$ y está en el primer plano en el tiempo $t-1$ es menos probable que permanezca en el primer plano (foreground) en el tiempo *t* a que regrese al fondo (background). Nótese que las transiciones BF y FB corresponden a eventos de oclusión y des-oclusión temporal, y que un píxel no puede cambiar de capa sin ir a través de un evento de oclusión.

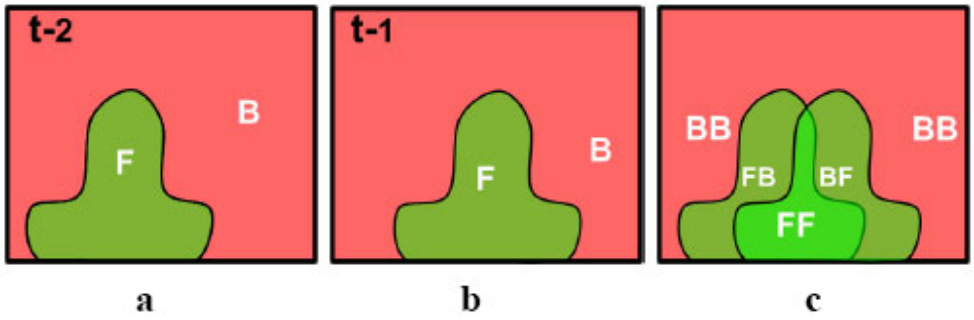

Figura 5.2. Transición temporal en un píxel. (a,b) Un objeto se mueve hacia la derecha de la trama $t-2$ a la trama $t-1$. (c) Entre los píxeles de las dos tramas el objeto puede permanecer en el fondo o en el primer plano o cambiar de una capa a otra.

En la siguiente tabla se muestran las transiciones temporales, las probabilidades del fondo (background) son el complemento de las probabilidades del primer plano (foreground).

α^{t-1}	α^{t-2}	$p\left(\alpha^t = F \middle\| \alpha^{t-1}, \alpha^{t-2}\right)$
F	F	β_{FF}
F	B	β_{FB}
B	F	β_{BF}
B	B	β_{BB}

Tabla 5.1. Información previa para transiciones temporales

Estas intuiciones se capturan probabilísticamente y se incorporan al diseño para la minimización de la energía mediante una cadena de Markov de segundo orden tal y como se muestra en la figura 5.1. La transición temporal previa se obtiene de los datos etiquetados y después son almacenados en una tabla como se muestra en la tabla 5.1. Observar que existen ocho (2^3) transiciones posibles, debido a la normalización probabilística $\left(p(\alpha^t = B \middle\| \alpha^{t-1}, \alpha^{t-2}) = 1 - p(\alpha^t = F \middle\| \alpha^{t-1}, \alpha^{t-2})\right)$ la tabla de información previa temporal tiene solamente cuatro grados de libertad representados por los cuatro parámetros $\beta_{FF}, \beta_{FB}, \beta_{BF}, \beta_{BB}$. Esto conduce al siguiente término temporal previo:

$$V^T\left(\alpha^t, \alpha^{t-1}, \alpha^{t-2}\right) = \eta \sum_n^N \left[-\log p(\alpha_n^t \middle| \alpha_n^{t-1}, \alpha_n^{t-2})\right] \qquad \textbf{(5. 8)}$$

Donde $\eta < 1$ es un factor de descuento para permitir la cuenta múltiple de los píxeles no-indepenientes. Como se explicará mas adelante, el valor óptimo de η (así como de otros parámetros del CRF) se debe probar para cada secuencia de video.

5.1.3. Energía espacial

Existe una tendencia natural para la segmentación de fronteras para alinearlas con los contornos de una imagen de alto contraste. Lo anterior se representa por un término de energía de la forma:

$$V^S(\alpha, z) = \gamma \sum_{(m,n)\in C} [\alpha_m \neq \alpha_n] \left(\frac{\varepsilon + e^{-\mu \|z_m - z_n\|^2}}{1+\varepsilon} \right) \qquad \textbf{(5. 9)}$$

Donde el subíndice (m,n) representa un par de píxeles vecinos. C es el conjunto de pares de píxeles vecinos. El parámetro de contraste μ se elige para que sea: $\mu = \left(2\left\langle \|z_m - z_n\|^2 \right\rangle\right)^{-1}$; donde $\langle\cdot\rangle$ denota la expectativa sobre todos los pares de vecinos en una muestra de una imagen. El término de energía $V(\alpha, z)$ representa una combinación de un *Ising*[1] previo para la coherencia de etiquetado junto con una probabilidad de contraste que actúa como descuento parcial de los términos de coherencia. El contraste γ es un parámetro de fuerza para la coherencia previa así como para la probabilidad de contraste. La constante ε es la constante de "*dilución"* para el contraste y es igual a cero para segmentación puramente de color. Sin embargo, varios experimentos recomiendan que el valor mas apropiado es $\varepsilon = 1$.

5.1.4. Probabilidad del color

El término $U^C(\cdot)$ en la ecuación (5.7) es el *log* de la probabilidad del color. En trabajos previos, las probabilidades del color han sido modeladas en términos de GMM (Gaussian Mixture Models) en RGB donde las mezclas del fondo (background) y del primer plano (foreground) se conocen por medio de la Maximización de la Expectativa (EM). Sin embargo, se han encontrado algunos problemas con la inicialización del algoritmo EM. En vez de usar EM, en este diseño se modelan las probabilidades del color del fondo y del primer plano de forma no paramétrica, como por ejemplo histogramas en el espacio de color YUV. El término de color $U^C(\cdot)$ se define como:

$$U^C(\alpha, z) = -\rho \sum_n^N \log p(z_n | \alpha_n) \qquad \textbf{(5. 10)}$$

Una normalización probabilística requiere que $\sum_z p(z|\alpha = F)$, y de forma similar para la probabilidad del fondo. Esta representación no paramétrica niega la necesidad de tener que fijar el número de componentes GMM así como tener que esperar para la convergencia del algoritmo EM.

[1] El término ising hace referencia al modelo matemático que se usa para modelar diversos fenómenos donde los bits de información interactúan en pares y producen efectos colectivos

El modelo de probabilidad de color del primer plano (foreground) se obtiene adaptativamente sobre tramas sucesivas, basado en la información del primer plano segmentado en la trama previa. Las probabilidades son después almacenadas en tablas *look-up* 3-D, construidas a partir de los histogramas del color. La distribución de color en el fondo se construye a partir de una observación extendida inicial del mismo. Después, la distribución es estática en el tiempo.

5.1.5. Probabilidad del movimiento

Para tratar con el movimiento, esto se puede realizar mediante el cálculo del flujo óptico. Sin embargo, un cálculo confiable del flujo es costoso desde el punto de vista computacional, además de que presenta algunas dificultades por el problema de apertura y la regularización. Dichos problemas pueden ser evitados modelando directamente las características normalmente utilizadas para obtener el flujo, conocidas como derivadas espacial y temporal $m = (g, \dot{z})$.

La probabilidad del movimiento por lo tanto, captura las características del fondo y del primer plano. Sin embargo, la naturaleza de éste modelo generativo sugiere un diseño para modelar la probabilidad del movimiento que debe captar información de la segmentación. Refiriéndonos a la figura 5.2, la segmentación de un píxel cae en una de las cuatro clases, FF, BB, FB, BF. En este algoritmo, se modelan las características del movimiento observado en la imagen $m_n^t = \left(g_n^t, z_n^t\right)$, en el tiempo *t* y para el píxel n, como condicionadas en las combinaciones de las etiquetas de segmentación α_n^{t-1} y α_n^t. Este es un modelo natural ya que la derivada temporal $\dot{z}_n^t$ se calcula para las tramas $t-1$ y t, por lo que esto dependerá de la segmentación de esas tramas. La distribución de BB refleja la relativa constancia del estado del fondo (background) y las derivadas temporales son pequeñas en magnitud. La distribución FF refleja un cambio temporal mayor y como es de esperarse, eso es algo relacionado con la magnitud del gradiente espacial. Las distribuciones de transición BF y FB muestran cambios temporales mayores ya que las muestras temporales en el tiempo $t-1$ y en el tiempo t indican una frontera del objeto.
Las probabilidades del movimiento son evaluadas como parte de la energía total en el término:

$$U^M\left(\alpha^t, \alpha^{t-1}, m^t\right) = -\sum_n \log p\left(m_n^t \middle| \alpha_n^t, \alpha_n^{t-1}\right) \qquad \textbf{(5. 11)}$$

6.1.6. Interferencia por la minimización de la energía

Desde un principio, el principal objetivo fue la maximización de la probabilidad a posteriori. Sin embargo, las restricciones de causalidad en los sistemas en tiempo real no lo permiten. Bajo causalidad y habiendo estimado $\hat{\alpha}^1, ..., \hat{\alpha}^{t-1}$, una forma simple de calcular $\hat{\alpha}^t$ sería de la siguiente manera:

$$\hat{\alpha}^t = \arg\min E\left(\alpha^t, \hat{\alpha}^{t-1}, \hat{\alpha}^{t-2}, z^t, m^t\right) \quad \textbf{(5. 12)}$$

Congelando todos los estimadores antes de generar t es un diseño extremo, además de que se obtienen mejores resultados reconociendo la variabilidad en al menos el paso inmediato previo. Por lo tanto, la energía dada por la ecuación (5.12) se reemplaza por la energía esperada dada por la siguiente ecuación:

$$\varepsilon_{\alpha^{t-1}\left|\hat{\alpha}^{t-1}\right.} E\left(\alpha^t, \alpha^{t-1}, \hat{\alpha}^{t-2}, z^t, m^t\right) \quad \textbf{(5. 13)}$$

Donde la densidad condicional para el tiempo $t-1$ se modela como:

$$p\left(\alpha^{t-1}\left|\hat{\alpha}^{t-1}\right.\right) = \prod_n p\left(\alpha_n^{t-1}\left|\hat{\alpha}_n^{t-1}\right.\right) \quad \textbf{(5. 14)}$$

y

$$p\left(\alpha^{t-1}\left|\hat{\alpha}^{t-1}\right.\right) = \nu + (1-\nu)\delta(\alpha^{t-1}, \hat{\alpha}^{t-1}) \quad \textbf{(5. 15)}$$

Y ν (donde, $\nu \in [0,1]$) es el grado al cual la segmentación binaria en el tiempo $t-1$ produce una distribución de la segmentación. En la práctica, con $\nu > 0$ (típicamente $\nu = 0.1$) se previenen errores en los estados del fondo y del primer plano.

Esta factorización de la distribución de segmentación a lo largo de los píxeles hace que el cálculo de la expectativa (5.13) sea completamente tratable. La alternativa de representar completamente la incertidumbre en la segmentación es computacionalmente muy costosa. Finalmente, la segmentación $\hat{\alpha}^t$ se calcula por el corte de grafo binario.

5.2. Implementación del algoritmo

Como se describió desde el inicio de este capítulo el algoritmo propuesto segmenta videos mediante una fusión probabilística de las señales de movimiento, color y contraste junto con información espacial y temporal. El modelo forma un Campo Aleatorio Condicional (CRF) y sus parámetros se prueban discriminativamente. La componente del movimiento del modelo evita el cálculo del flujo óptico y en vez de éste, usa un novedoso y efectivo modelo de probabilidad basado en derivadas espacio-temporales y condicionado a pares de tramas.

La coherencia espacio-temporal se explota mediante una energía sensible de contraste combinada con una cadena de Markov temporal de segundo orden. En

términos de eficiencia, éste algoritmo se compara favorablemente con respecto a las técnicas *stereo* en tiempo real existentes.
Para la implementación del algoritmo se requiere de una función de energía compleja que depende tanto de los datos de la imagen como de las etiquetas de segmentación. La energía codifica los términos que hacen cumplir la coherencia de la segmentación en el tiempo (tramas consecutivas) y en el espacio (píxeles cercanos) así como los datos de la imagen. Se requiere de participación del usuario en la elección de los objetos a segmentar por lo que es un algoritmo que segmenta objetos de un video de una forma supervisada. La segmentación es estimada mediante la minimización de la función de energía anteriormente mencionada.

El algoritmo realiza una segmentación en dos capas (background/foreground) y separa el fondo del primer plano de un video. En el algoritmo se realiza la minimización de la energía compuesta por la suma de los cuatro términos anteriormente explicados, V^T, V^S, U^C, U^M.

5.3. Resultados Experimentales

El algoritmo se probó con 6 diferentes secuencias de video monoculares, cada una de ellas con diferentes características, Para la secuencia de video 1 (AC) se puede observar que solamente se encuentra un objeto en movimiento y el fondo estático. La segmentación de dos capas para este caso arroja buenos resultados, en la figura 5.3, trama 17 se observa que se pierden algunos píxeles sobre todo en las primeras tramas cuando el foreground esta casi estático o con poco movimiento, en la figura 5.3, trama 33 se observa que una vez que el objetos se mueve la segmentación obtiene fronteras mas exactas del objeto.

Figura 5.3. Secuencia 1 (AC), Trama de video (background/foreground) y resultados de la segmentación, trama 17, 33 y 160.

Para esta secuencia se utilizaron los siguientes valores para los parámetros:

$\eta = 0.4; \quad \gamma = 0.4 \quad \varepsilon = 1$

$\rho = 0.8 \quad \nu = 0.1$

En al figura 5.4 se muestra una de las tramas del video y tres tramas tomadas del video segmentado. Esta segunda secuencia de video aplicada al programa posee tres objetos en movimiento, sin embargo en las tramas de video que se muestran solo aparecen dos de ellos, en pocas tramas se pueden ver los tres objetos y para este caso los resultados no son tan buenos como cuando se tiene solamente un objeto a segmentar en el primer plano (foreground). Hay que recordar que el programa requiere un etiquetado del color a mano al inicio, dicho etiquetado se realiza delineando los contornos o fronteras de los objetos que pertenecen al foreground (primer plano) y que son los que se espera sean segmentados por el algoritmo, sin embargo en este diseño intervienen otra información como es la información del movimiento.

El algoritmo identifica los objetos que no tienen un etiquetado inicial, sin embargo las fronteras de los mismos no son tan bien definidas como aquellos que si fueron etiquetados a mano. En la secuencia anterior la mujer es la que pertenece al etiquetado inicial y las personas que se mueven detrás de ella no por lo que se observa que éstos últimos no están muy bien delineados.

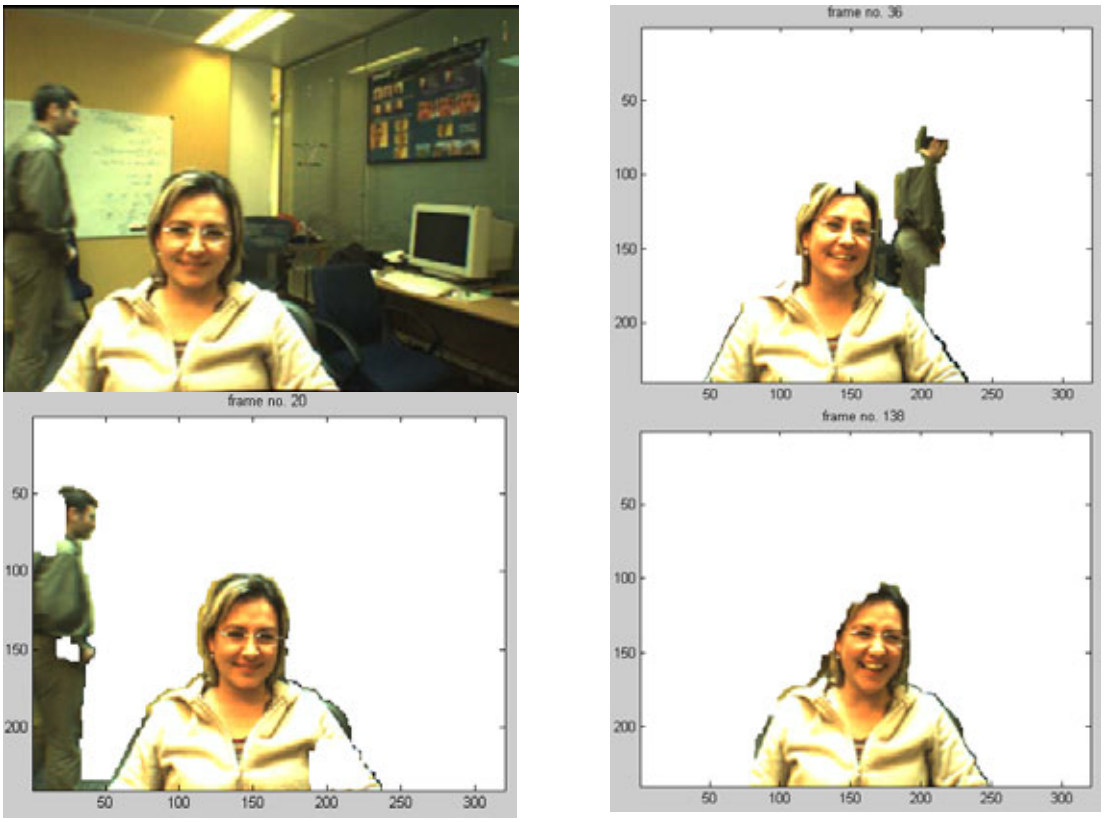

Figura 5.4. Secuencia 2 (IU). Trama del video y resultados de la segmentación de video (tramas 20, 36 y 138)

Los valores de los parámetros con los que se obtuvieron los mejores resultados de la segmentación para esta secuencia son los siguientes:

$$\eta = 0.25; \quad \gamma = 0.4 \quad \varepsilon = 1$$
$$\rho = 0.5 \quad \nu = 0.1$$

En la figura 5.5 se muestran los resultados obtenidos por el algoritmo para la segmentación de la secuencia video digital 3 (JM). Para dicha secuencia el algoritmo presenta buenos resultados cuando el objeto que pertenece al primer plano (foreground) no tiene movimientos abruptos, una vez que el objeto cambia de un estado de poco movimiento a otro con mucho movimiento, se pierden algunos píxeles sobre todo de las partes con mayor luminosidad como son la frente del hombre sentado (ver trama 158 de la figura 5.5). Además después del movimiento abrupto se crean ciertas oclusiones perdiendo en cierta forma las fronteras que definen al objeto, si el objeto continúa moviéndose, estas fronteras se recuperan pero en aquellas partes donde el objeto ya no registra movimiento las fronteras ya no son recuperadas. Se puede observar que en la etapa final del video (trama 442 de la figura 5.5) se recupera casi totalmente las fronteras del objeto.

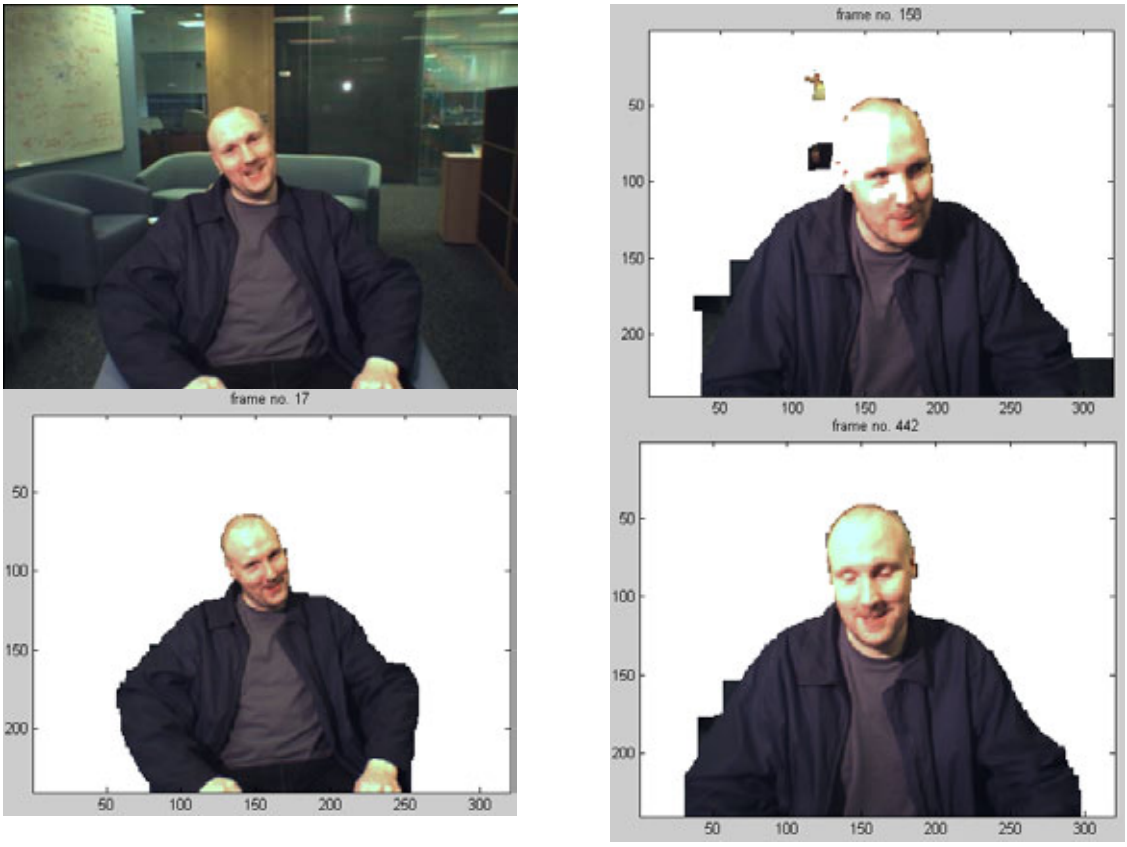

Figura 5.5. Secuencia 3 (JM). Trama del video y resultados de la segmentación de video (trama 17, 158 y 442).

Los valores de los parámetros para esta secuencia son los siguientes:

$\eta = 0.4; \quad \gamma = 0.4 \quad \varepsilon = 1$

$\rho = 0.9 \quad \nu = 0.1$

En la figura 5.6 se muestran los resultados de la segmentación de la secuencia de video 4 (VK). Se observa que se obtienen buenos resultados de segmentación en aquellas partes donde el contraste entre el fondo y objeto en movimiento es mayor ya que en las regiones donde el foreground y el fondo tienen el mismo color existen problemas para definir las fronteras del objeto perteneciente al foreground (ver hombro superior derecho del hombre en las secuencias 150 y 210).

Por otra parte el algoritmo sigue eficientemente el movimiento de la persona sobre todo cuando ésta levanta los brazos, sin embargo las fronteras no se perciben completamente definidas.

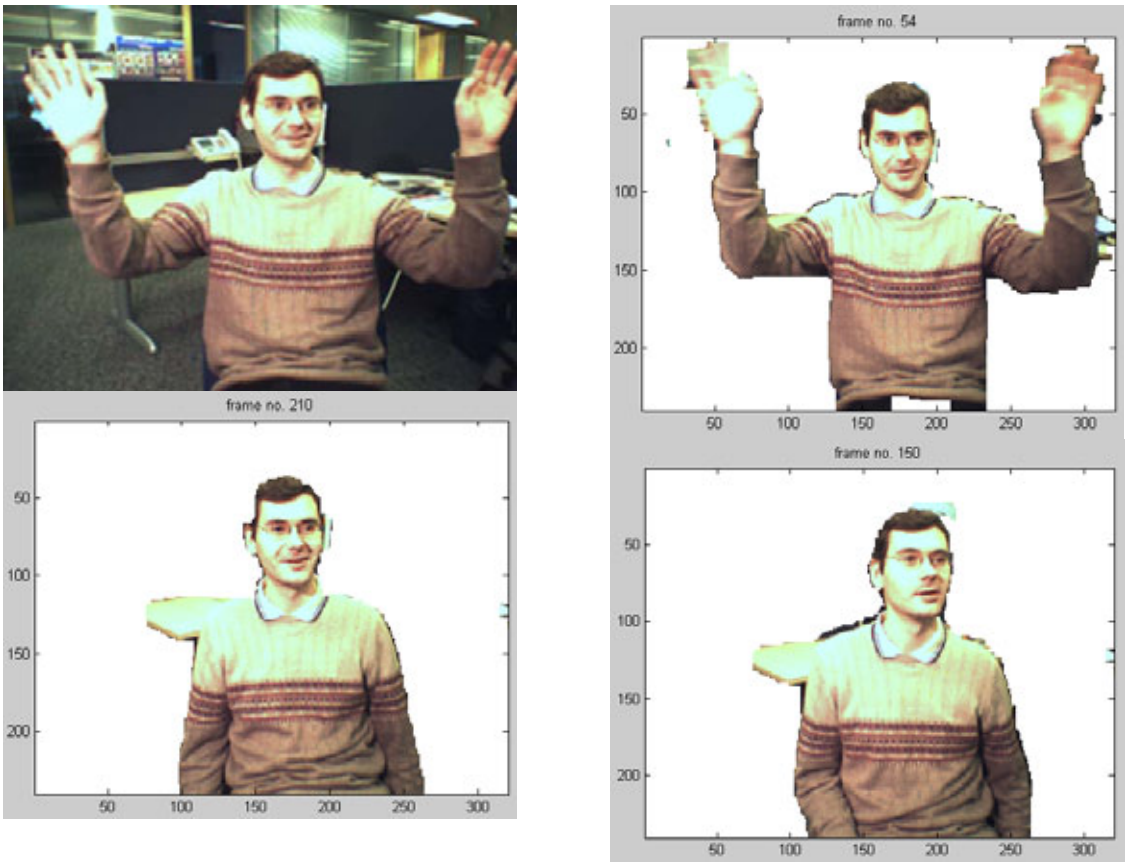

Figura 5.6. Secuencia 4 (VK). Trama de la secuencia de video y resultados de la segmentación de video en dos capas (tramas 54, 150 y 210)

Los valores de los parámetros para esta secuencia son los siguientes:

$\eta = 0.4; \quad \gamma = 0.4 \quad \varepsilon = 0.5$

$\rho = 0.8 \quad \nu = 0.1$

La figura 5.7 muestra los resultados de la segmentación de la secuencia de video 5 (MS) donde se observa un solo objeto en movimiento perteneciente al primer plano (foreground). La segmentación proporciona buenos resultados y presenta problemas en la definición de las fronteras del objeto en movimiento en aquellas zonas donde entre el fondo y el objeto no existe contraste o tienen color semejante (ver suéter oscuro del hombre sentado y fondo oscuro alrededor de la zona). Al inicio el algoritmo no define bien las fronteras sobre todo en la parte de la cabeza debido a que el objeto esta prácticamente estático, sin embargo, una vez que comienza a moverse los resultados de la segmentación mejoran considerablemente.

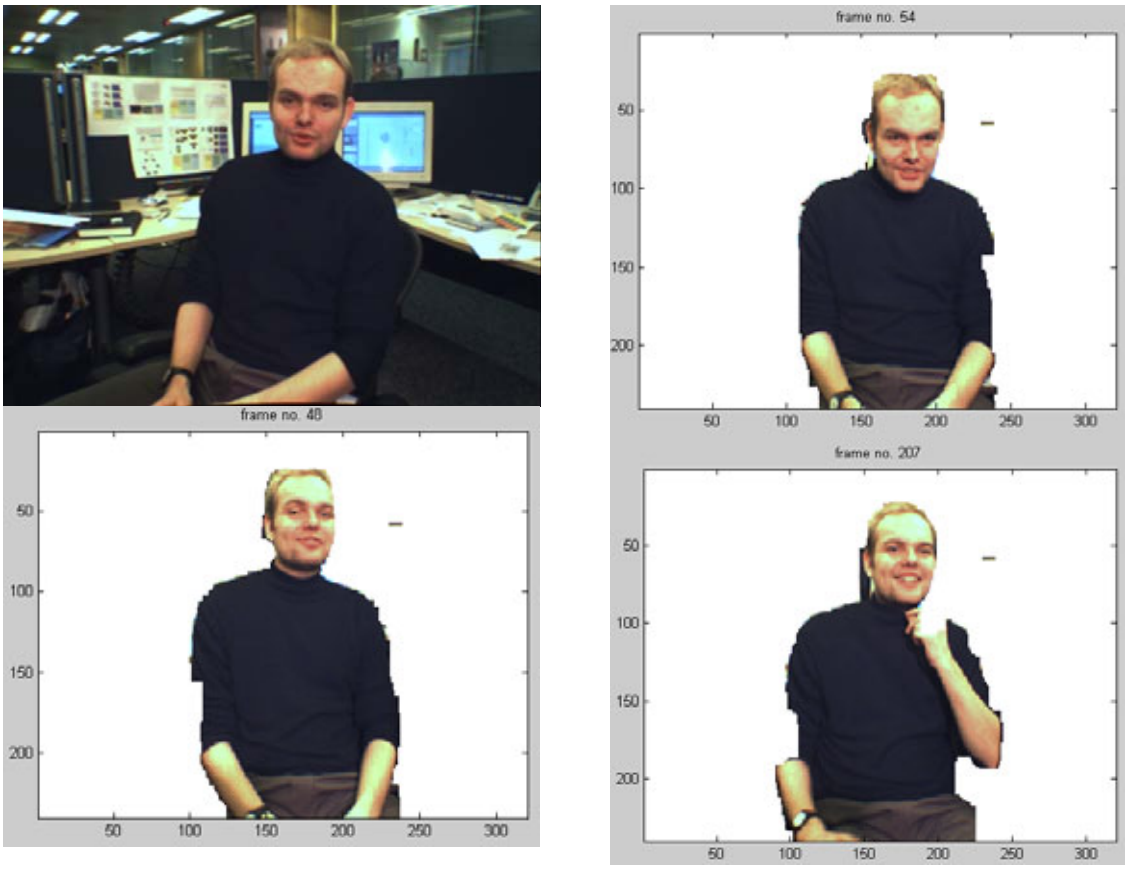

Figura 5.7. Secuencia 5 (MS). Trama del video antes de la segmentación y resultados de la segmentación de dos capas (tramas 48, 54 y 207)

Los valores de los parámetros para esta secuencia son los siguientes:

$$\eta = 0.1; \quad \gamma = 0.1 \quad \varepsilon = 1$$
$$\rho = 0.75 \quad \nu = 0.1$$

En la figura 5.8 se presentan los resultados de la segmentación de la secuencia de video (IUJW) donde en un momento existen hasta 4 objetos en movimiento dentro de la escena, los 4 objetos en movimiento no aparecen en todo el video lo cual nos permite distinguir las diferencias de la segmentación cuando hay pocos objetos y cuando hay varios objetos moviéndose. Para el etiquetado de color inicial solo se realiza en las dos personas sentadas, observando una mejor definición de las fronteras en dichos objetos, detrás de ellos aparecen personas moviéndose rápido, para este caso el algoritmo es capaz de detectar el movimiento incluso en aquellos objetos que no han recibido un etiquetado previo a mano pero la definición para las fronteras de dichos objetos no presentan buenos resultados en cuanto a la exactitud.

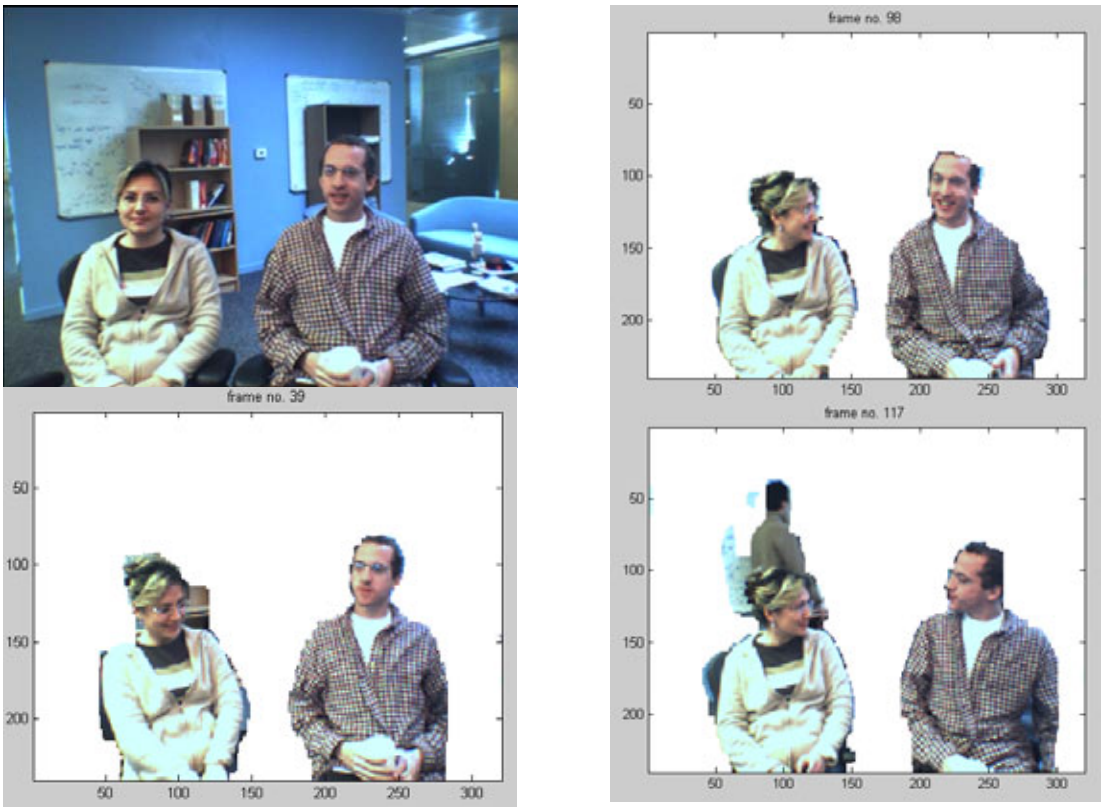

Figura 5.8. Secuencia 6 (IUJW). Trama de la secuencia del video a segmentar y resultados de la segmentación de dos capas (tramas 39, 98 y 117).

Los valores de los parámetros para esta secuencia son los siguientes:

$\eta = 0.1; \quad \gamma = 0.4 \quad \varepsilon = 1$

$\rho = 0.7 \quad \nu = 0.1$

Hay que recordar que el modelo presentado en este capítulo requiere para la segmentación de secuencias de video a color de la minimización de una ecuación de energía. La ecuación de la energía se compone de cuatro términos, V^T , V^S, U^C, U^M, explicados a detalle al inicio de este capítulo

Los valores de los parámetros $(\eta, \rho, \varepsilon, \gamma)$ dependen de la secuencia a segmentar, cada uno de éstos afecta cada término dentro de la ecuación de energía por lo que deben ajustarse de forma empírica.

η es el término que le da peso al término temporal previo V^T, γ es una constante que le de fuerza a la continuidad temporal y a la probabilidad del contraste, ε es una constante de dilución para el contraste y en la mayoría de los casos con un valor de $\varepsilon = 1$ se obtienen buenos resultados. ρ es un parámetro que le da fuerza al término de probabilidad del color (U^C).

En el término de la probabilidad del movimiento U^M se calculan 4 probabilidades que son las siguientes: BB, BF, FF, FB. Cuando un píxel que pertenece al fondo (B) en la trama *t-2* y en la trama *t-1* pertenece al Foreground (F), es más probable que en el la trama *t* regrese al fondo (B) por lo que debe dársele más peso a esta transición, para hacer esto se agrega un factor que aumenta dicha probabilidad, este factor depende de la secuencia de video a segmentar y típicamente esta entre 1.1 y 1.6. Cabe mencionar que este factor es diferente para cada una de las secuencias de video mostradas.

5.4. Análisis de resultados

En el capítulo 5 de este trabajo se presentó el diseño, implementación y resultados de un diseño que segmenta secuencias de video a color en dos capas, haciendo diferencia entre el fondo (background) y el primer plano (foreground).

Para dicho algoritmo implementado en MATLAB se realizaron pruebas con 6 secuencias de video diferentes. Se observó que para secuencias de movimiento con un objeto en movimiento en el primer plano (*foreground*) se obtienen buenos resultados sobre todo cuando el objeto empieza a tener movimiento. En términos generales se obtienen resultados buenos en la definición de las fronteras. En secuencias con mayor movimiento donde se involucra movimiento de brazos por ejemplo, el algoritmo es eficiente en la detección del movimiento, sin embargo las fronteras y contornos no se definen con la misma exactitud.

Por otra parte en secuencias de video que tienen poca diferencia de color entre el fondo y el objeto a segmentar, es decir, en aquellas donde existe poco contraste en los resultados de la segmentación se pierden algunos píxeles sin embargo los parámetros que controlan el peso de cada uno de los cuatro elementos de la función de energía pueden ajustarse manualmente de tal forma de obtener mejores resultados. Estos parámetros se fijaron de forma empírica para cada secuencia de video probada.

El algoritmo presentado en el capítulo 5 segmenta secuencias de video a color, no requiere del cálculo del flujo óptico, sin embargo para la parte de inicialización requiere un etiquetado de color a mano que consiste con base en una trama del video identificar el número de objetos pertenecientes al primer plano, es decir, aquellos objetos que se desea segmentar y después se delinean los contornos de los objetos con el ratón de la computadora, con esta información introducida por el usuario, el algoritmo obtiene resultados de una segmentación de dos capas prácticamente en tiempo real con aplicaciones en video conferencia y en post-procesamiento de video digital. Por lo anterior, se clasifica a este método como semi-automatico o supervisado. El usuario puede extraer los objetos pertenecientes al primer plano y colocarlos en un fondo diferente al del video original. Para ver los resultados de las seis diferentes secuencias referirse al capítulo 4.

El diseño del capítulo 4 requiere una carga computacional mayor que el descrito en el capítulo 5, además de que sólo aplica para secuencias de video en blanco y negro, dicho algoritmo para la etapa de inicialización no requiere que el usuario proporcione información de ningún tipo y la segmentación se realiza de forma completamente automática. Al ser un diseño automático y con gran carga computacional, el tiempo de procesamiento se eleva. En este trabajo no se implementó dicho algoritmo, solamente se presentaron los resultados obtenidos por Yang Wang *et al.* [51]

CONCLUSIONES

Conclusiones

Se presentó el diseño y la implementación de un método para la segmentación espacio-temporal de un video digital. La segmentación se realiza en dos capas, diferenciando los objetos del *foreground* (definido por el usuario) del fondo (*background*). El modelo consiste en una fusión probabilística de las señales de movimiento, color y contraste junto con información espacial y temporal, la componente del movimiento del modelo evita el cálculo del flujo óptico y en vez de éste usa un novedoso y efectivo modelo de probabilidad basado en derivadas espacio – temporales.

Se realizaron pruebas con 6 secuencias de video diferentes y el algoritmo arrojó buenos resultados. Los valores de algunos parámetros con los que se controla cada uno de los cuatro términos de la ecuación de energía se fijaron de forma empírica y se ajustaron para cada secuencia individual teniendo valores diferentes para cada una de ellas. Los resultados de este algoritmo dependen en cierta forma de la secuencia de video a segmentar. Una de las características de este diseño es que realiza el procesamiento prácticamente en tiempo real. Las posibles aplicaciones para este diseño son en video conferencia y post-procesamiento de video digital.

En este trabajo también se presentaron varios métodos para la segmentación de un video digital basada en objetos así como algunos criterios para su clasificación. En el capítulo 4 se presentó un diseño propuesto por Yang Wang *et al.* [51], el cual se basa en una estructura probabilística para lograr una segmentación espacio-temporal de secuencias de video. En este diseño se hace interactuar la información de movimiento, la información de la segmentación de intensidad y la información de la segmentación de video por medio de una red bayesiana haciendo uso de tres tramas, tratando de este modo con el problema de las oclusiones.

Este tipo de modelo es una extensión de la técnica presentada en el capítulo 4 donde tanto el campo de vectores de movimiento como el campo de la segmentación de video se calculan de forma simultánea agregando además un campo para la segmentación de intensidad con lo que se logra tener una segmentación de video con fronteras ó bordes de los objetos más definidos.

El modelo se compone de diseños de fusión y división de regiones haciendo interactuar la información espacial (intensidad) con la información temporal (movimiento) llegando a resultados exitosos para la localización y segmentación de objetos en movimiento en un video. Este es un diseño robusto de segmentación de video que requiere una carga computacional grande. Además cabe mencionar que es un diseño para una segmentación no supervisada y para secuencias de video monocromáticas.

El procedimiento utilizado en el diseño de segmentación de video presentado en el la sección 4.4.1 obtiene la solución estimando el MAP [51] con un procedimiento de optimización que maximiza de forma iterativa la densidad de probabilidad condicional de los tres campos. Las restricciones espacio-temporales se formulan con campos aleatorios de Markov [2]. Este método demuestra ser robusto y coherente espacio-temporalmente.

Se pudo observar que los resultados a los que se llegó en el trabajo de Yang Wang *et al.* [51] se definen de forma exacta las fronteras de los objetos además de que identifica de forma eficiente el movimiento, sin embargo la implementación de dicho algoritmo es mas compleja y requiere mayor carga computacional, lo cual se traduce en un mayor tiempo de procesamiento que no hace posible el uso de dicho modelo para aplicaciones en tiempo real. Por otro lado este algoritmo fue diseñado para secuencias de video monocromáticas y no aplica para videos a color como. Una de las ventajas de éste algoritmo es que es un algoritmo no supervisado por lo que realiza la segmentación de forma automática.

En cuanto al algoritmo propuesto, éste presenta ventajas ya que la poca carga computacional en parte debido al uso de las derivadas espacio-temporales en vez del cálculo del campo de vectores de movimiento, como se realiza en la mayoría de los diseños, hace posible un tiempo de procesamiento bastante corto permitiendo poder usarlo en aplicaciones en tiempo real. Sin embargo los resultados del modelo propuesto dependen en gran medida del tipo de secuencias de video y del número de objetos en movimiento. Por lo que se obtienen mejores resultados para secuencias de video con pocos objetos en movimiento, además de que este diseño no trata con las oclusiones como se realiza en el modelo del capítulo 4 el cual se basa en un diseño de tres tramas.
Además el diseño propuesto necesita un etiquetado inicial realizado sobre una de las tramas del video donde se requiere la entrada mediante el ratón de la computadora de los objetos a segmentar.

En cuanto al trabajo a futuro y la forma de mejorar el modelo propuesto, éstos se concentran en encontrar un procedimiento eficiente para el cálculo óptimo de los parámetros que controlan el peso de cada uno de los términos de la ecuación de energía para no hacerlo de forma manual y empírica. Por otro lado extender este diseño de dos tramas a un diseño de tres tramas que permita tratar con el problema de las oclusiones y finalmente hacer lo anterior sin sacrificar el corto tiempo de procesamiento el cual sin duda fue uno de los principales logros del algoritmo.

BIBLIOGRAFÍA

Bibliografía

[1] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. “Bilayer Segmentation of Live video”. CVPR 2006.

[2] A. Murat Tekalp. *Digital Video Processing*. New Jersey, Prentice Hall PTR, 1995.

[3] Ahmed Elgammal, Ramani Duraiswami, David Harwood y Larry S. Davis. Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance. Proceedings of the IEEE, July 2002.

[4] B.D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision”, Proc. DARPA Image Understanding Workshop, pp. 121-130, 1981.

[5] B. K. P. Horn and B. G. Schunck, “Determining optical flow”, Artif. Intell., vol. 17, pp. 185-203, 1981.

[6] Bruce D. Lucas y Takeo Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision.* Proceedings of the 1981 DARPA Image Understanding Workshop, April 1981.

[7] C. Stiller, “Object-oriented video coding employing dense motion fields”, Proc. Int. Conf. ASSP, Adelaide, Australia, April 1994.

[8] Christopher K. Eveland, Kurt Konolige y Robert C. Bolles. *Background Modeling for Segmentation of Video-Rate Stereo Sequences.* Computer Vision and Pattern Recognition, 1998.

[9] D. J. Heeger, “Model for the extraction of image flow”, J. Opt. Soc. Am. A, vol. 4, No. 8, pp. 1455-1471, 1987.

[10] D. W. Murray and B.F. Buxton, “Scene segmentation from visual motion using global optimization”, IEEE Trans. Patt. Anal. Mach. Intel., vol. 9, No. 2, pp. 220-228, Mar 1987.

[11] Dengsheng Zhang y Guojun Lu. *Segmentation of Moving Objects in Image Sequence: A Review*. Circuits, Systems and Signal Processing, 2001

[12] E. Dubois and J. Honrad, “Estimation of 2-D motion fields from image sequences with application to motion-compensated processing”, in Motion Analysis and Image Sequence Processing, M. I.Sezan and R. L. Langendijk, eds., Norwell, MA: Kluwer, 1993.

[13] Ebroul Izquierdo. *Disparity/Segmentation Analysis: Matching with Adaptive Windows and Depth Driven Segmentation.* IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video Processing for Emerging Interactive Multimedia Services, June 1999.

[14] F. Moscheni, S. Bhattacharjee, and M. Kunt, *"Spatiotemporal segmentation based on region merging"*, IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, No.5, pp.897 – 915, May 1998.

[15] G. Adiv, "Determining three-dimensional motion and structure from optical flor generated by several moving objects", IEEE Trans. Pattern Anal. Mach. Intell., vol. 7, pp. 384-401, 1985.

[16] G. Gordon, T. Darrell, M. Harville y J. Wood_ll. *Background Estimation and Removal Based on Range and Color.* Proceedings of the Computer Vision and Pattern Recognition, vol. 2, June 1999.

[17] G. Sullivan, "Multi-hypothesis motion compensation for low bit-rate video coding", Proc. IEEE Int. Conf. ASSP, Minneapolis, MN, vol. 5, 1993.

[18] H. H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results", Artificial Intelligence, vol. 33, pp. 299-324, 1987.

[19] Hayit Greenspan, Jacob Goldberger y Arnaldo Mayer. *Probabilistic Space Time Video Modeling via Piecewise GMM*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004.

[20] I. Patras, E. A. Hendriks, and R. L. Lagendijk, "Video segmentation by MAP labeling of watershed segments", IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, No. 2, pp. 326 – 332, Feb. 2001.

[21] J. L. Barron, D. J. Fleet y S. S. Beauchemin. *Performance of Optical Flow Techniques*. International Journal of Computer Vision.

[22] John Y. A. Wang y Edward H. Adelson. *Representing Moving Images with Layers.* IEEE Transactions on Image Processing, September 1994.

[23] Kiran Challapali, Tomas Brodsky, Yun-Ting Lin, Yong Yan y Richard Yi Chen. *Real-time object segmentation and coding for selective-quality video communications*. IEEE Transactions on Circuits and Systems for Video Technology, June 2004.

[24] L. Jacobson and H. Wechsler, “Derivation of optical flow using a spatio-temporal frequency approach”, Comp. Vision Graph. Image Proc., vol. 38, pp. 57-61, 1987.

[25] L. Lucchese y S. K. Mitra. *Color image segmentation: a state of the art survey.* Proceedings of the Indian National Science Academy (INSA-A), vol. 67. March 2001.

[26] M. Hoetter and R. Thoma, “Image segmentation based on object oriented mapping parameters estimation”, Signal Proc., vol. 15, pp. 315 – 334, 1988.

[27] M. M. Chang, A.M. Tekalp, and M.I. Sezan, “Motion field segmentation using an adaptive MAP criterion”, Proc. Int. Conf. ASSP, Minneapolis, MN, April 1993.

[28] M.M. Chang, M. I. Sezan, and A.M. Tekalp, “An algorithm for simultaneous motion estimation and scene segmentation”, Proc. Int. Conf. ASSP, Adelaide, Australia, April 1994.

[29] M. M. Chang, A. M. Tekalp, and M. I. Sezan, *“Simultaneous motion estimation and segmentation”*, IEEE Trans. Images Process., vol. 6, No. 8, pp. 1326-1333, Aug. 1997.

[30] Mario Figueiredo y Anil Jain. Unsupervised Learning of Finite Mixture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, March 2002.

[31] Mark Everingham y Barry Thomas. *Supervised Segmentation and Tracking of Nonrigid Objects Using a Mixture of Histograms Model*. IEEE International Conference on Image Processing, vol. 1, October 2001.

[32] Michael J. Black y Allan D. Jepson. *Estimating optical flow in segmented images using variable-order parametric models with local deformations*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996.

[33] Michael J. Black y P. Anandan. *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*. Computer Vision and Image Understanding, January 1996.

[34] Michal Irani, Benny Rousso y Shmuel Peleg. *Computing Occluding and Transparent Motions*. International Journal of Computer Vision, February 1994.

[35] Milan Sonka, Vaclav Hlavac y Roger Boyle. Image Processing: Analysis and Machine Vision, 2nd edition. Thomson-Engineering, September 1998.

[36] Noel Brady y Noel O'Connor. *Object Detection and Tracking Using an EM-Based Motion Estimation and Segmentation Framework.* Proceedings of the 8th IEEE International Conference on Image Processing, vol.1, September 1996.

[37] P. Anandan, J.R. Bergen, k.j. Hanna, and R. Hingorani, *"Hierarchical model-based motion estimation"* in Motion Analysis and Image Sequence Processing, M.I. Sezan and R. L. Lagendijk, eds., Norwell, MA: Kluer, 1993.

[38] P.B. Chou and C.M. Brown, "The theory and practice of Bayesian image sequences", Int. J. Comp. Vision, vol. 4, pp. 185-210, 1990.

[39] P. Wayne Power y Johann A. Schoonees. *Understanding Background Mixture Models for Foreground Segmentation*. Proceedings of Image and Vision Computing, New Zeland, 2002

[40] Remi Megret y Daniel DeMenthon. A Survey of Spatio-Temporal Grouping Techniques. University of Maryland, College Park, 2002.

[41] Richard O. Duda, Meter E. Hart y David G, Store. *Pattern Classification*, 2a edición. New York, Wiley-Interscience, November 2001.

[42] Roberta Piroddi y Theodore Vlachos. *Multiple-Feature Spatiotemporal Segmentation of Moving Sequences using a Rule-based Approach*. British Machine Vision Conference, pp. 353-362, 2002

[43] Roberto Castagno, Touradj Ebrahimi y Murat Kunt. *Video Segmentation based on Multiple Features for Interactive Multimedia Applications*. IEEE Transactions on Image Processing, September 1998.

[44] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception", Biol, Cybern., vol. 60, pp. 79-97, 1988.

[45] Simon Baker y Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, pp. 221-255, 2004.

[46] Sohaib Khan y Mubarak Shah. *Object Based Segmentation of Video Using Color, Motion and Spatial Information*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, December 2001.

[47] Stephen J. McKenna, Yogesh Raja y Shaogang Gong. *Tracking colour objects using adaptive mixture models.* Image and Vision Computing, 1999.

[48] T. N. Papps, An adaptive clustering algorithm for image segmentation, *IEEE Trans. Image Process.*, vol. 4, no. 5, pp. 901–914, May 1992.

[49] Y. H. Yang y M. D. Levine. *The Background Primal Sketch: An Approach for Tracking Moving Objects.* Machine Vision Application, 1992.

[50] Yaakov Tsaig and Amir Averbuch. *Automatic Segmentation of Moving Objects in Video Sequences: A Region Labeling Approach*. Circuits and Systems for Video Technology, IEEE Transactions on, July 2002.

[51] Yang Wang, Kia-Fock, Tele Tan y Jian-Kang Wu. *Spatiotemporal Video Segmentation Based on Graphical Models.* IEEE Transactions on image processing, vol. 4, No. 7, July 2005.

[52] Yu-Pao Tsai, Chih-Chuan Lai, Yi-Ping Hung y Zen-Chung Shih. *A Bayesian Approach to Video Object Segmentation via Merging 3-D Watershed Volumes*. IEEE Transactions on Circuits and Systems for Video Technology, 2005.

Páginas web consultadas:

[53]http://www.depeca.uah.es/docencia/doctorado/cursos04_05/82854/docus/CursoVision7.pdf

[54] http://www.dfmf.uned.es/actividades/no_reglada/laboratorio/segmentacion1.pdf

[55] http://www.lfcia.org/~cipenedo/cursos/Ip/Tema6/nodo6_2.html

[56] http://lfcia.org/~cipenedo/cursos/Ip/Tema7/nodo7_2.html

[57] http://www-gth.die.upm.es/~macias/rep/0607/docs0607/SimulatedAnnealing-JMG-Doctorado0607-v3.pdf

[58] http://www.cbsr.ia.ac.cn/users/szli/mrf_book/Chapter_8/node135.html

[59] http://www.cbsr.ia.ac.cn/users/szli/MRF_Book/Chapter_8/node141.html

ANEXO I

Anexo I. Campos aleatorios de Markov y campos de Gibbs

I.1. Definiciones generales

Los campos aleatorios van a proporcionar medidas de probabilidad sobre un dominio de definición que tenga relaciones de tipo espacial o temporal. El conjunto de posiciones donde se define el campo va a denominarse malla o rejilla y se va a denotar por el conjunto S. En la textura el conjunto S representa el conjunto de píxeles en una estructura matricial (2D), como puede verse en la figura I.1(a). En el contorno el conjunto S representa cada uno de los radios trazados desde el centro en una estructura lineal (1D), como puede verse en la figura I.1(b).

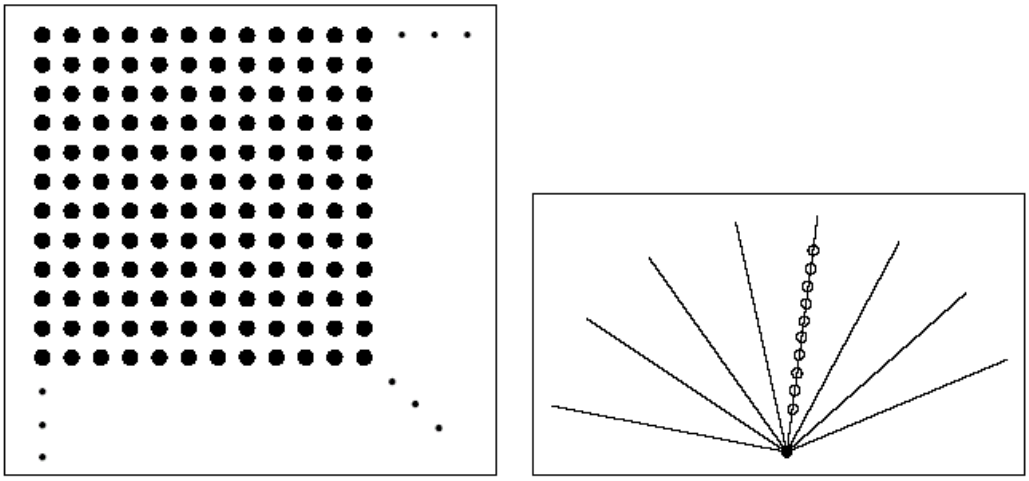

Figura I.1. Ejemplo de rejilla (a) textura y (b) contorno

Para cada posición $s \in S$ se va a definir un espacio de estados Λ_s. Para la textura, s es un píxel y el espacio de estados Λ_s para cada píxel corresponde a los niveles de gris. Para el contorno, s es un radio y el espacio de estados Λ_s para cada radio corresponde a cada uno de los puntos del radio por donde puede pasar el contorno. En la figura I.1(b), éstos corresponden a los círculos blancos a lo largo del radio. Con $x_s \in \Lambda_s$ vamos a denotar un valor de gris del píxel o una posición para el radio correspondiente. $\Omega = (\Lambda_s)_{s \in S}$ es el espacio de todas las configuraciones o contornos posibles que se pueden definir en S. Por otro lado, con $x = (x_s)_{s \in S}$ se denota una configuración en particular, es decir una textura o un contorno. Se supone que todos los espacios y configuraciones son finitos. Si todos los píxeles tienen el mismo rango de niveles de gris para el caso de la textura y el mismo rango de posiciones para todos los radios en el caso del contorno, se dice que el espacio Ω es homogéneo.

Una vez claro el dominio de definición del campo, falta definir una medida de probabilidad o distribución de probabilidad en el espacio de configuraciones Ω. A cada configuración (textura o contorno) $x \in \Omega$ se le asigna una probabilidad $\Pi(x) \geq 0$, tal que:

$$\sum_{x \in \Omega} \Pi(x) = 1 \qquad \textbf{(I.1)}$$

Un suceso $E \subset \Omega$ corresponderá a un conjunto de configuraciones. Será un conjunto de texturas o un conjunto de contornos para los que se define su probabilidad simplemente como:

$$\Pi(E) = \sum_{x \in E} \Pi(x) \qquad \textbf{(I. 2)}$$

Igualmente, se pueden definir subconjuntos $\mathrm{A} \subset S$ del espacio de definición S. En el caso de la textura, corresponderá a una subimagen y en el caso del contorno, a una porción de éste. $\Omega_{\mathrm{A}} = (\Lambda_s)_{s \in A}$ corresponde al espacio de configuraciones de la subimagen o porción de contorno. $x_{\mathrm{A}} = (x_s)_{s \in \mathrm{A}}$ corresponde a una subimagen o trozo de contorno en particular. La probabilidad en este subespacio se define utilizando el concepto clásico de probabilidad marginal. Se va a denotar con X, X_{A} ó X_s las variables aleatorias correspondientes a los espacios Ω, Ω_A ó Λ_s respectivamente. X es la variable aleatoria que representa a la textura o el contorno, X_{A} la que representa a una subimagen de la textura o una porción del contorno y X_s la que representa a un píxel de la textura o aun radio del contorno. Se dice que un campo definido en la rejilla S con espacio de configuraciones Ω y medida de probabilidad asociada Π es un campo aleatorio o estocástico si para todo $x \in \Omega$ se cumple que $\Pi(x) > 0$, es decir, si la distribución o medida de probabilidad cumple la condición de positividad. Por lo tanto, si el modelo de textura es un campo aleatorio, significa que todas las imágenes o configuraciones son posibles (con mayor o menor probabilidad). En el caso del contorno, de la misma forma, significa que todos los contornos van a ser posibles.

I.2. Características locales, vecindarios y cliques[1].

Para un campo aleatorio se puede definir un tipo de probabilidad condicionada denominada característica local del campo, definida para $A \subset S$ como

$$\Pi(X_{\mathrm{A}} = x_{\mathrm{A}} / X_{S|\mathrm{A}} = x_{S|\mathrm{A}}) \qquad \textbf{(I. 3)}$$

[1] Entiéndase por clique como un sub-conjunto de individuos pertenecientes a un grupo mas grande que están mejor identificados entre ellos que con el resto del grupo, en este caso clique es un sub-conjunto de píxeles.

Las características locales siempre están definidas gracias a la propiedad de positividad de los campos aleatorios. En el ejemplo de la textura, sería la probabilidad de que una sub-imagen de la textura tome un cierto valor, condicionada al resto de la textura. En el caso del contorno, sería la probabilidad de que una porción del contorno tome un cierto valor, condicionada al resto del mismo. Las dependencias en S van a ser, en general, locales. Esto quiere decir que en una textura un píxel va a depender de los píxeles cercanos y que en un contorno el valor de éste en un radio va depender del valor del contorno en los radios cercanos. Por esta razón se va a definir para cada posición $s \in S$ un conjunto $\partial(s) \subset S$, que corresponde a las posiciones de S de las que s depende. Los elementos $\partial(s)$ se denominan vecinos de s. La colección de conjuntos $\partial = \{\partial(s) : s \in S\}$ se denomina sistema de vecindario o vecindario de la rejilla S. Un sistema de vecindario debe cumplir dos propiedades:

- Que s no sea vecino de sí mismo
$$s \notin \partial(s) \qquad \textbf{(I. 4)}$$
- Que si s es vecino de t, éste último lo sea del primero
$$s \notin \partial(t) \Leftrightarrow t \in \partial(s) \qquad \textbf{(I. 5)}$$

En general, los sistemas de vecindario son homogéneos. Esto quiere decir que conociendo los vecinos de una posición s se puede conocer cuáles son los vecinos de otra posición t sin más que desplazar a t el sistema de vecinos de s. Se puede decir que son invariantes en el espacio. Los sistemas de vecinos pueden ser también isotrópicos. Esto quiere decir que se comportan de la misma forma en todas las direcciones (son invariantes a rotaciones en las direcciones principales de la rejilla).

En el caso del modelo de textura, sea $s = (i, j)$ y $t = (k, l)$ dos píxeles de la imagen tal que $t \in \partial(s)$; se define el orden c del vecindario (para el caso homogéneo e istrópico) como el menor entero que cumpla:

$$c \geq (k - i)^2 + (l - j)^2 \qquad \textbf{(I. 6)}$$

para todos los vecinos $t = (k, l)$ de $s = (i, j)$. En la figura I.1 podemos ver vecindarios de los órdenes más comunes: 1, 2, 4, 5 y 8, para el caso de la textura. El punto negro representa el píxel s y el punto blanco cada uno de los píxeles t vecinos de s. Puesto que la imagen definida en la rejilla S tiene dimensiones finitas, los vecindarios de los píxeles cerca del borde de la imagen no pueden ser iguales que los vecindarios de los píxeles interiores. El concepto de homogeneidad no se puede cumplir para los píxeles de borde; sin embargo, esto siempre va a ocurrir y se va a segur diciendo que el vecindario es homogéneo suponiendo implícitamente el efecto de bordes. Hay que recordar que $\partial(s) \subset S$ para todo $s \in S$, como se ha mencionado.

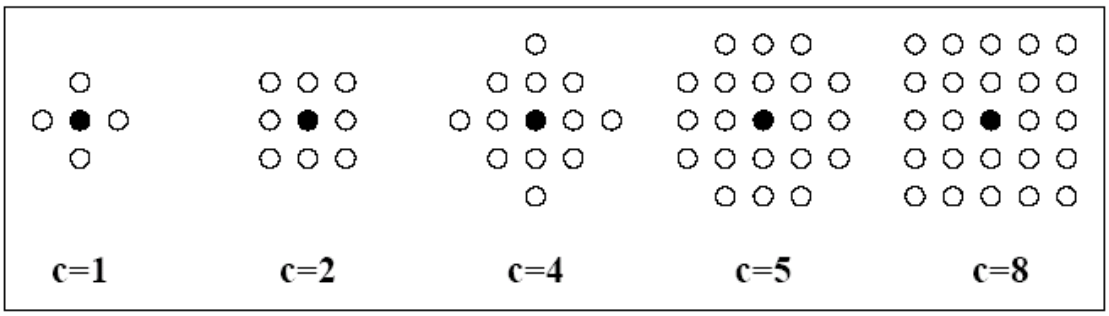

Figura I.2. Vecindarios de órdenes 1, 2, 4, 5 y 8

En el caso del modelo de contorno, si s y t son radios tales que $t \in \partial(s)$, se define el orden del vecindario (para el caso homogéneo e isotrópico) como el menor entero que cumpla:

$$c \geq |s - t| \qquad \textbf{(I. 7)}$$

para todos los vecinos t de s. En la figura I.4 se pueden ver vecindarios de los órdenes más comunes: 1, 2 y 3 para el caso del contorno. Se ha representado cada radio mediante un punto. Puesto que el contorno es cerrado, no se tienen problemas con los bordes y se puede tener un vecindario homogéneo en sentido estricto que cumpla que $\partial(s) \subset S$ para todo $s \in S$.

Dado un sistema de vecindario ∂ en S, se dice que un subconjunto $C \subset S$ es un clique, si dos elementos cualesquiera de C (diferentes entre sí) son vecinos. El conjunto de todos los cliques de S se denotará con C. En el caso de vecindarios homogéneos los cliques se van a poder clasificar en tipos, según la relación espacial entre los elementos que lo forman

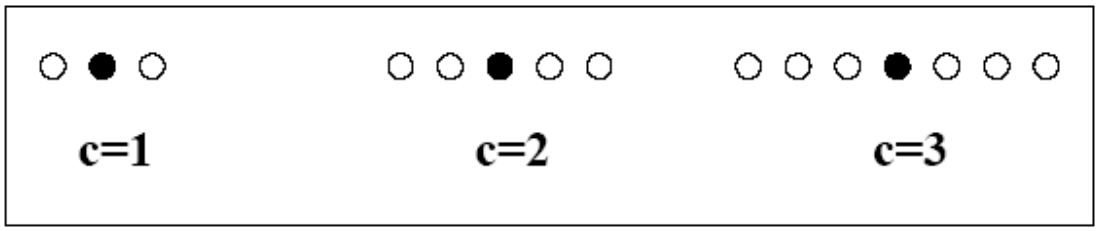

Figura I.3. Vecindarios de órdenes 1, 2 y 3 para el modelo de contorno

Las figuras I.5 y I.6 representan los tipos de cliques definibles a partir de los vecindarios de órdenes 1 y 2, respectivamente, a partir del modelo de textura. Los cliques de dos elementos para orden cinco se pueden ver en la figura I.7. En esta figura el círculo negro representa un elemento que pertenece al clique y el círculo blanco un elemento que no pertenece al clique. Los cliques de dos elementos se denominan cliques de pares.

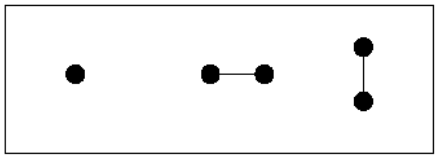

Figura I.4. Tipos de cliques para el vecindario de orden 1 en el modelo de textura

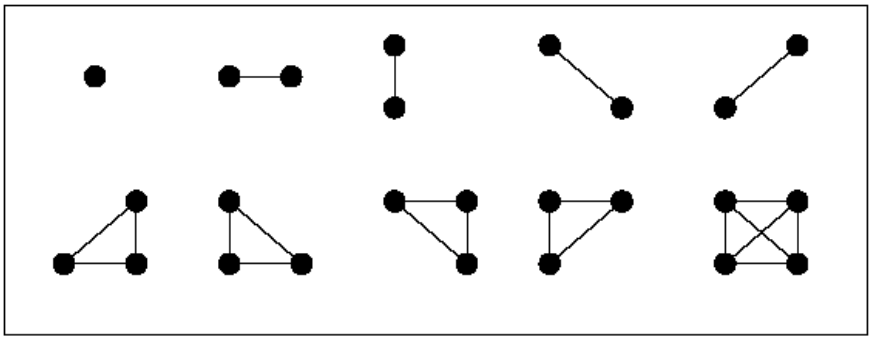

Figura I.5. Tipos de cliques de pares para el vecindario de orden 2 en el modelo de textura

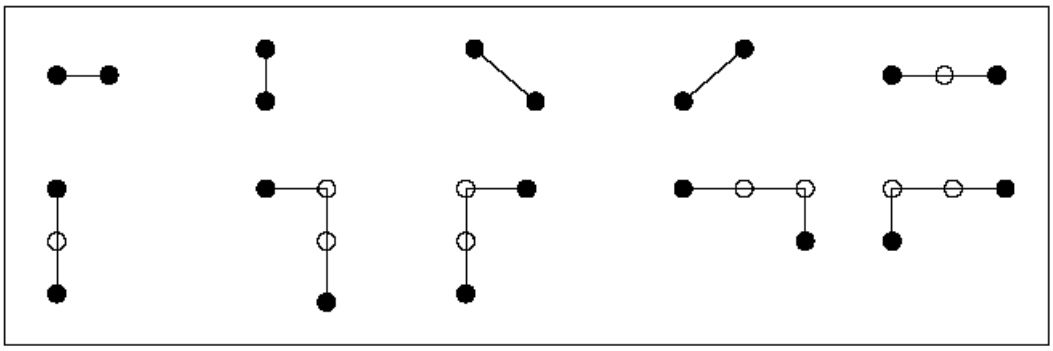

Figura I.6. Tipos de cliques de pares para el vecindario de orden 5 en el modelo de textura

Con respecto al modelo de contorno, en la figura I.8 se pueden ver los tipos de cliques para el vecindario de orden uno y en la figura I.9 para el de orden dos. Se dice que un campo aleatorio es de Markov o MRF con respecto al vecindario ∂ si para todo $x \in \Omega$:

$$\Pi(X_s = x_s | X_r = x_r, r \neq s) = \Pi(X_s = x_s | X_r = x_r, r \in \partial(s)) \qquad \textbf{(I. 8)}$$

con $s, r \in S$,

I.3. Campos de Gibbs, funciones de energía y potenciales

Se dice que un campo es de Gibbs o GRF (*Gibbs Random Field* por sus siglas en inglés) si la medida o distribución de probabilidad se puede poner de la forma:

$$\Pi(x) = \frac{\exp(-H(x))}{Z} \qquad \textbf{(I. 9)}$$

donde *H* es la función de energía que induce el campo y *Z* es la función de partición dada por:

$$Z = \sum_{z \in \Omega} \exp(-H(z)) \qquad \textbf{(I. 10)}$$

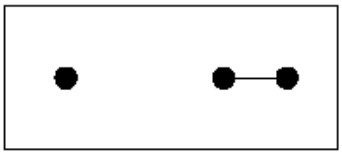

Figura I.7. Tipos de cliques para el vecindario de orden uno en el modelo de contorno

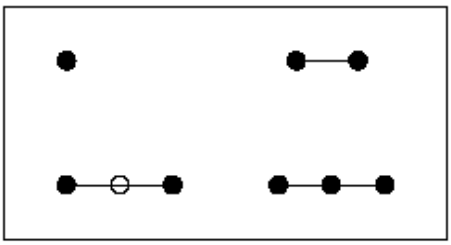

Figura I.8. Tipos de cliques para el vecindario de orden 2 en el modelo de contorno

Si una medida o distribución de probabilidad se puede poner como la ecuación I.9 se dice que es una medida o distribución de Gibbs.
La función energética *H* esta definida sobre el espacio de configuraciones Ω. En la práctica, debido al tamaño de este espacio, se trabaja en el dominio de las características locales. Para poder separar las diferentes interacciones locales presentes globalmente en la función de energía, se definirá una familia de funciones con dependencias locales $\{U_A : A \subset S\}$ en Ω. A esta familia se le denomina potencial u y a cada función elemento función potencial. Se tiene una función potencial U_A para cada subconjunto $A \subset S$. El número de funciones potenciales es igual al número de subconjuntos posibles de S, que es $2^{|S|}$.

Para que un potencial u induzca un GRF, ésta debe cumplir dos condiciones

$$U_0 = 0 \qquad \textbf{(I. 11)}$$

$$U_A(x) = U_A(y) \Leftrightarrow x_A = y_A \qquad \textbf{(I. 12)}$$

La ecuación I.11 indica que la función potencial para el conjunto vacío es la función nula y la ecuación I.12 indica que la función potencial U_{A} sólo va a depender de la configuración en $\mathrm{A} \subset S$, es decir, se podría poner $U_A(x) = U_A(x_A)$. En base a ellas, la energía del GRF se calcula como:

$$H_u = \sum_{A \subset S} U_A \qquad \textbf{(I. 13)}$$

Se dice que el GRF viene inducido por el potencial u siempre que la energía del campo H_u venga dada por la ecuación I.13. La forma de estas funciones potenciales puede ser todo lo arbitrario que se desee.

En el caso de que las dependencias sean locales y dadas por un sistema de vecindario ∂, el número de funciones potenciales no nulas se reduce considerablemente. De hecho de todos los subconjuntos $\mathrm{A} \subset S$ sólo es necesario considerar unos pocos, concretamente, los dados por el conjunto de cliques c. Esto simplifica considerablemente la definición de un GRF utilizando funciones potenciales.

Dado un sistema de vecindario ∂, un potencial u es un potencial de vecindario con respecto a ∂ si $U_A = 0$, siempre que A no sea un clique, es decir, siempre que $A \notin C$. Por lo tanto si C es un clique, las funciones potenciales no nulas se van a denotar por U_C. La energía del campo se calcula ahora como:

$$H_u(x) = \sum_{C \in c} U_C(x_C) = \sum_{C \in c} U_C(x_C) \qquad \textbf{(I. 14)}$$

Esta energía induce un GRF denominado de vecindario para el potencial u con respecto a ∂. Para vecindarios homogéneos, en general, el potencial va a ser homogéneo. Esto en la práctica significa que en lugar de una función diferente para cada clique $C \in c$ se tendrá una función potencial para cada tipo de clique. Por ejemplo, para le modelo de textura con vecindario de orden 2 con 8 vecinos (figura I.5), se tiene 10 clases de cliques, entonces si tanto el vecindario como el potencial son homogéneos, se tendrá como máximo 10 funciones potenciales, mientras que el número de cliques de c es un número mucho mayor. En general, aunque el sistema de vecindario sea isótropo, las funciones potenciales no lo van a ser.

En el caso en que las funciones potenciales U_A sean nulas para $|A| > 2$, es decir, para conjuntos de más de dos elementos, el potencial u se denomina potencial de pares. Si éste viene referido a un cierto vecindario ∂ entonces además A debe ser un clique. En este caso, se dice que el potencial u es un potencial de pares referido a ∂.

Dado un sistema de vecindario ∂ se verá cual es la relación entre MRFs y GRFs. Esta relación se conoce como *el teorema de Hammersley y Clifford* y establece que un campo aleatorio es un MRF con respecto al vecindario ∂ si y sólo si es un GRF para ∂. Es decir, *son totalmente equivalentes*.

Dado un campo de Markov para un vecindario ∂, debe existir un potencial de vecindario u que induzca dicho campo. Las expresiones para estas funciones potenciales a partir del MRF son bastante complicadas pero teóricamente demostrables. Un GRF para un vecindario ∂ inducido por el potencial de vecindario u, es un MRF para ese vecindario.

En general, la relación entre un GRF o un MRF y un potencial no es unívoca. De hecho dos potenciales diferentes pueden inducir el mismo MRF. Para evitar esta ambigüedad, el potencial se normaliza con respecto a una configuración fijada por convenio, de forma que todas las funciones potenciales para esa configuración sean nulas. De esta forma si se puede decir que la relación entre un GRF y un potencial normalizado es unívoca. Esta configuración que anula todas las funciones potenciales se suele denominar configuración de *referencia* o de *vacío*. Puesto que la energía se determina como la suma global de funciones potenciales, la energía de la configuración de referencia también es cero. Se dice que esta función energética inducida está normalizada.

La relación entre un GRF y la energía normalizada también es unívoca. La ecuación I.8 que definía un MRF a partir de las características locales para cada $s \in S$ con respecto al sistema de vecindario ∂ se puede determinar a partir del potencial u como:

$$\Pi(X_A = x_A \left| X_{S|A} = x_{S|A}\right.) = \Pi(X_A = x_A \left| X_s = x_s, s \in \partial(A)\right.)$$
$$= \frac{\exp\left(- \sum_{C \in c, C \cap A \neq 0} U_C(x) \right)}{\sum_{z_A \in \Omega_A} \exp\left(- \sum_{C \in c, C \cap A \neq 0} U_C(z_A X_{S|A}) \right)} \qquad \textbf{(I. 15)}$$

donde se define el vecindario para cada subconjunto $A \subset S$ como:

$$\partial(A) = \left\{ \bigcup_{s \in A} \partial(s) \right\} \Big| A. \qquad \textbf{(I. 16)}$$

Para finalizar se puede repetir la ecuación I.9 usando el potencial de vecindario u

$$\Pi(x) = \frac{\exp\left(-\sum_{C \in c} U_C(x) \right)}{Z} \qquad \textbf{(I. 17)}$$

donde Z es la función de partición que en función del potencial u se puede determinar como:

$$Z = \sum_{z \in \Omega} \exp\left(-\sum_{C \in c} U_C(z) \right) \qquad \textbf{(I. 18)}$$