



Centro de Investigación en Matemáticas

Modelos de campos de medida Markovianos cuadráticos con control de entropía para segmentación de imágenes

Tesis

que para obtener el grado de

Maestro en ciencias

con especialidad en Ciencias de la Computación

presenta:

Jesús Omar Ocegueda González

Directores de tesis

Dr. Mariano Rivera Meraz

Dr. José Luis Marroquín Zaleta

Guanajuato, Gto., 14 de Agosto de 2006

Índice general

1. Introducción	1
2. Representación	4
2.1. Definiciones preliminares	4
2.2. Campos Aleatorios Markovianos	7
2.3. Campos aleatorios de Gibbs	8
2.4. Estimación Bayesiana	10
3. Problemas clásicos de procesamiento de imágenes de bajo nivel	13
3.1. Restauración	13
3.1.1. Potenciales robustos	15
3.2. Estimación de la disparidad estereoscópica	20
3.3. Estimación de flujo óptico	21
3.4. Segmentación	23
4. Modelos utilizados en el problema de segmentación	28
4.1. Cadenas de Markov - Monte Carlo (MCMC)	28
4.2. Campos de medida de Gauss-Markov	33
4.3. Campos Ocultos Markovianos de Medida (HMMF)	34
5. Campos de Medida Markovianos cuadráticos con control de entropía	37
5.1. Derivación teórica	37
5.2. Método propuesto de optimización con restricciones	40
5.3. Campos de Medida de Gauss-Markov con entropía controlada (EC-GMMF)	44

6. Resultados experimentales	46
6.1. Graph-cuts	48
6.2. Comparación	52
7. Aplicaciones	61
7.1. Splines de producto tensorial	62
7.2. IRM sintéticas	64
7.2.1. Introduciendo un modelo adicional	67
7.2.2. Considerando unicamente la región del cerebro	67
7.3. IRM reales (conjunto de datos de Harvard)	68
7.3.1. Usando solamente los voxeles del cerebro	70
7.3.2. Usando los datos completos	70
7.3.3. Tiempo de ejecución	73
7.4. Discusión	75
7.5. Estimación del flujo óptico	78
7.5.1. Discusión	83
7.6. Restauración	84
7.7. Segmentación de imágenes provenientes de colposcopías	87
7.7.1. Trabajo previo	88
7.7.2. Aplicación de EC-QMMF para series de tiempo	92
7.7.3. Posibles modificaciones y extensiones	94
7.7.4. Discusión	96
8. Resumen y conclusiones	100
Bibliography	102

Agradecimientos

Quiero expresar mi gratitud al Dr. Mariano Rivera, el Dr. José Luis Marroquín y el Dr. Johan Van Horebeek por sus recomendaciones para mejorar la calidad de este trabajo de tesis y por el apoyo que me han brindado a lo largo de toda mi estancia en la comunidad CIMAT-FAMAT. Sus conocimientos y su experiencia, no solo en el aspecto profesional sino también en lo personal, y sobre todo su manera tan amena de transmitirlos han hecho de estos últimos años una experiencia invaluable para mí. Ahora que pienso en las pláticas, discusiones, clases, las “sesiones de regaños” y los momentos compartidos dentro y fuera de la comunidad, no puedo menos que decir gracias por esos momentos, por esos consejos, por esas enseñanzas y sobre por su amistad.

Quiero agradecer también a mis amigos y compañeros dentro y fuera del CIMAT, desde aquellos que conozco desde hace más de dos años y con quienes he seguido conviviendo hasta la fecha como a quienes recién conocí ahora en la maestría. En especial quisiera agradecer a Zazil, por estar conmigo en muchos momentos difíciles, a Paty, y Óscar por esas charlas tan padres y divertidas, a Poncho, César, Carlos, Raúl y su esposa Tania, Eugenio y a mis amigos de la FAMAT, Kuko, Limo, Pável, Pepe, Neto y los “niños” de la Olimpiada de Informática que me aguantaron como uno de sus entrenadores y que me dieron tantas satisfacciones, sobre todo por las tardes de convivencia en la facultad. A mis amigos del equipo *Mandelbrot* Marte, Edgar y Fory con quienes he vivido experiencias inolvidables y que a pesar de seguir caminos diferentes seguimos manteniendo una gran amistad.

Quisiera agradecer a la comunidad CIMAT-FAMAT en general por el ambiente tan agradable y estimulante y al CONACyT por el apoyo económico que recibí y sin el cual me habría sido imposible realizar mis estudios de maestría en el CIMAT.

Le dirijo un agradecimiento especial a los doctores Gabriel Acosta y Nicandro Cruz de la Universidad Veracruzana por invitarme a participar en el proyecto de segmentación de imágenes de colposcopia y

por su invitación a las instalaciones de la Maestría en Inteligencia Artificial.

Gracias a mi familia, Yeni, Sergio, Mary y en especial a mi madre por el ejemplo de lucha y sacrificio que me ha inspirado siempre a seguir adelante. Gracias a Azucena por permanecer en la distancia pero nunca en la ausencia.

Capítulo 1

Introducción

*Nobody can be successful
unless he loves his work.*

David Sarnoff.

No es sorprendente la magnitud del desarrollo que ha tenido el área del procesamiento digital de imágenes en los últimos años, si pensamos también en la gran cantidad de aplicaciones que de ésta ha surgido. Una imagen es útil en cualquier situación en la que se requiera tomar mediciones de “algo” a lo largo de una región (ya sea superficie o volumen). En una fotografía, por ejemplo, se toman mediciones de la cantidad de luz que un objeto emite o refleja. Existe una gran variedad de tipos de fotografías que se pueden obtener variando el tipo de filtro y el tipo de sensores que se utilicen en la captura, por ejemplo termografías (Fig. 1.1 a)) en la banda infrarroja del espectro, las populares fotografías (Fig. 1.1 b)), en la banda visible y las radiografías (Fig. 1.1 c)). Un campo de estudio que utiliza a fondo esta variedad de imágenes en diferentes bandas del espectro electromagnético es el campo de la *percepción remota* (*Remote sensing*), que estudia las imágenes de satélite (Fig. 1.1 d)). Analizando en conjunto todas estas imágenes tomadas en una misma región es posible determinar el tipo de suelo presente en esta región así como los diferentes elementos que pueden estar presentes en ella como puede ser la presencia o ausencia de vegetación, si se trata de una zona urbana, etc.

Un campo de aplicación especialmente interesante es el de las imágenes médicas, una de las herramientas más revolucionarias son las imágenes resonancia magnética (Figura 1.2), que permiten ver el interior del cuerpo humano sin necesidad de intervenciones quirúrgicas. En muchos casos es difícil para los expertos analizar las imágenes por lo que se han desarrollado herramientas para ayudar en el diagnóstico, con esto se abren nuevos campos de investigación en procesamiento de imágenes.

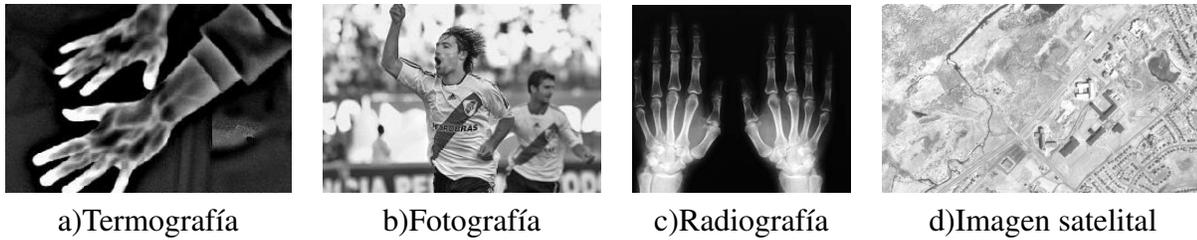


Figura 1.1: Fotografías tomadas a diferentes longitudes de onda.

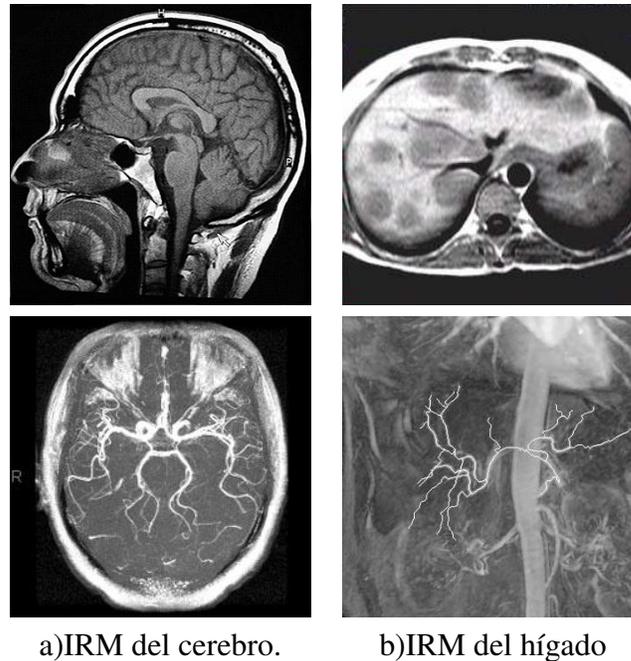


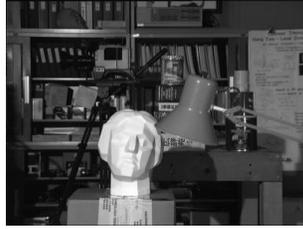
Figura 1.2: Imágenes de resonancia magnética.Fila 2: con medio de contraste (angiografía)

Otra area interesante del procesamiento de imágenes es el procesamiento de video o secuencias de imágenes (Figura 1.3), un caso particular es el procesamiento de *pares estéreo*. Un par estereo es un par de imágenes tomadas de la misma escena pero desde una posición y un ángulo ligeramente diferentes (para definiciones más precisas ver [1]). Con este tipo de imágenes y conociendo las condiciones en las que se tomaron las imágenes es posible determinar la distancia a la que se encuentran los diferentes objetos en la escena e incluso parte de su geometría.

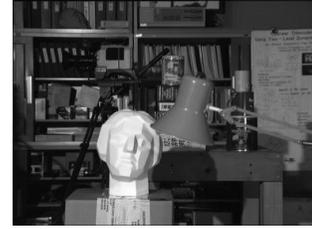
Por si solas, las imágenes son solo datos, organizados de una manera conveniente. La manera de explotar estos datos es la que define su aplicación. El “arte” del procesamiento de imágenes consiste en extraer información relevante de estos datos para que posteriormente esta información pueda ser utilizada ya sea por personas o bien por aplicaciones de más alto nivel. El “producto final” es una tecnología que



a)Secuencia de video.



b)Imagen izquierda de par estereo.



b)Imagen derecha de par estereo.

Figura 1.3: Secuencias de imágenes

resuelva algún problema del “mundo real”.

Capítulo 2

Representación

There is no true interpretations of anything; interpretation is a vehicle in the service of human comprehension. The value of interpretation is enabling others to fruitfully think about an idea.

Andreas Buja.

Con el objetivo de resolver *problemas* de procesamiento de imágenes necesitamos definir *modelos* que nos permitan plantear dichos problemas de una manera concisa, para después poder utilizar herramientas matemáticas para encontrar soluciones eficientes.

2.1. Definiciones preliminares

Lo primero que necesitamos es una manera de representar una imagen. La representación que utilizaremos será a través de un conjunto \mathcal{S} de *sitios*, cada uno de los cuales representa un *pixel* de la imagen. Dicho conjunto debe tener una estructura (los pixeles de las imágenes no están dispersos sino que están distribuidos regularmente), para modelar dicha estructura necesitamos describir la manera como los sitios se relacionan entre sí y esto lo hacemos definiendo un sistema de vecindades \mathcal{N} , asociado al conjunto de sitios \mathcal{S} como

$$\mathcal{N} = \{\mathcal{N}_r \mid r \in \mathcal{S}\}$$

donde \mathcal{N}_r es el conjunto de sitios de \mathcal{S} que son *vecinos* de r . La relación de vecindad debe ser simétrica y no-reflexiva, es decir

- $r \notin \mathcal{N}_r$
- $r \in \mathcal{N}_s \Leftrightarrow s \in \mathcal{N}_r$

Definición: un subconjunto $c \subset \mathcal{S}$ es *completo* si cada par de sitios $x, y \in c$ son vecinos.

Definición: un *clique* es un conjunto que contiene a un solo sitio o bien, un conjunto completo.

Para ilustrar las definiciones anteriores, consideremos el grafo de la la figura 2.1, en este grafo, los nodos representan sitios y la presencia de una arista entre los nodos x y y representa que x y y son vecinos.

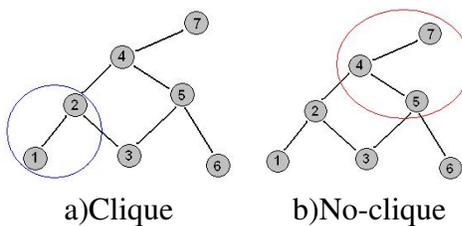


Figura 2.1: En a) ningún nodo que agreguemos será vecino simultáneamente de 1 y 2. En b) 5 no es vecino de 7.

Observemos que en el caso de grafos, la relación de vecindad debe definirse para cada par de sitios (nodos), por lo que esta es la clase más general de sistemas de vecindades que podemos tener.

Las relaciones entre sitios que se presentan en las imágenes es mucho más sencillo que las relaciones de grafos en general. De hecho podemos definir el sistema de vecindades **localmente** debido a que en el caso de las imágenes tenemos la noción de *cercanía* entre cualesquiera dos sitios. Dado un sitio $r \in \mathcal{S}$ podemos por ejemplo definir su vecindad como

$$\mathcal{N}_r = \{s \in \mathcal{S} \mid 0 < d(r, s) \leq \rho\}$$

donde d es la distancia euclídeana. En la figura 2.2 se muestran ejemplos para $\rho = 1$, $\rho = \sqrt{2}$ (suponiendo que la distancia de aristas verticales y horizontales es 1).

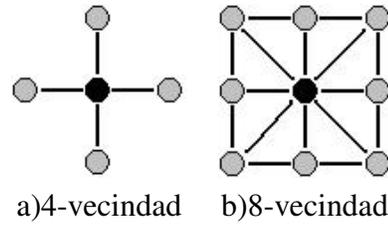


Figura 2.2: Dos tipos de vecindades definidas localmente. Los nodos en gris son los vecinos del nodo en negro.

Una vez definida una vecindad localmente, hemos definido todo el sistema de vecindades. En la figura 2.3 se muestra la representación (como grafo) de los sistemas de vecindades para todo el conjunto de sitios \mathcal{S}

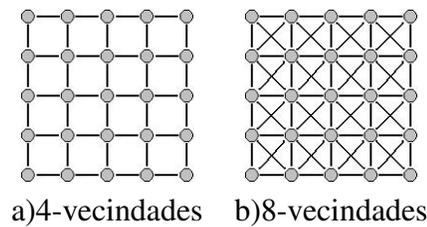


Figura 2.3: Sistemas de vecindades obtenidos al extender la definición de la vecindad local.

Otra ventaja de definir el sistema de vecindades a partir de una vecindad local, es que es fácil encontrar todos los *cliques*, basta observar la forma que estos tienen en la vecindad local. En la figura 2.4 se ilustra la forma que tendrán todos los cliques asociados a los sistemas de vecindades 4-conectados y 8-conectados.

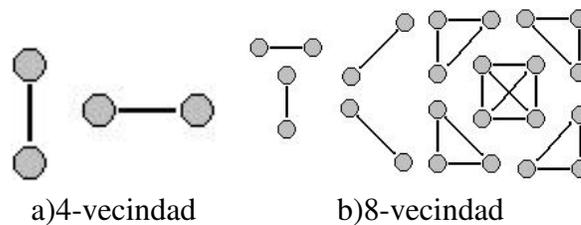


Figura 2.4: Cliques asociados a los sistemas de vecindades 4-conectados y 8-conectados.

El *orden* de un *clique* $c \subset \mathcal{S}$ es su cardinalidad $\#c$, y denotamos

$$\mathcal{C}_1 = \{\{r\} \mid r \in \mathcal{S}\}$$

$$\mathcal{C}_n = \{\{r_1, r_2, \dots, r_n\} \mid r_i \in \mathcal{N}_{r_j}, \forall i \neq j\}$$

Denotaremos por \mathcal{C} al conjunto de todos los conjuntos de cliques.

2.2. Campos Aleatorios Markovianos

Los elementos que definen una imagen, no son solamente el conjunto de sitios que la componen y la relación que hay entre ellos, sino también el **valor observado** en cada uno de los sitios. En una imagen en blanco y negro, por ejemplo, cada sitio en \mathcal{S} toma un valor que corresponde al nivel de gris de la imagen en el sitio correspondiente, en una imagen a color cada sitio toma tres valores que corresponden a la intensidad de los canales rojo, verde y azul (r,g,b). De esta manera, al haber definido el conjunto de sitios \mathcal{S} y su sistema de vecindades \mathcal{N} asociado, podemos pensar que solamente hemos definido el *dominio* o *región de definición* de una imagen. Ahora, para describir los valores que toma cada sitio $r \in \mathcal{S}$, podemos definir una función que asocie a cada sitio $r \in \mathcal{S}$ un elemento del conjunto de todos los posibles valores que puede tomar un sitio.

Definición: una *configuración* es una función

$$g : \mathcal{S} \rightarrow \mathcal{V}$$

donde \mathcal{V} es el conjunto de todos los valores posibles que podemos observar en cada sitio.

Denotaré por \mathbb{F} al conjunto de todas las posibles configuraciones definidas sobre \mathcal{S} para el conjunto de valores \mathcal{V} .

Por otro lado, si asociamos a cada sitio de $r \in \mathcal{S}$ una variable aleatoria F_r y consideramos el valor observado ($g(r)$) en el sitio r como una *realización* de dicha variable aleatorio, entonces podemos modelar una imagen como un *campo aleatorio*

$$F = \{F_r \mid r \in \mathcal{S}\}$$

y aprovechando la estructura de \mathcal{S} dada por \mathcal{N} , podemos modelar las dependencias de dichas variables aleatorias a través del sistema de vecindades.

Definición: un campo aleatorio F definido en \mathcal{S} es *Markoviano* respecto al sistema de vecindades \mathcal{N} si

$$P(f) > 0 \quad \forall f \in \mathbb{F}$$

$$P(F_r | F_{\mathcal{S} \setminus r}) = P(F_r | F_{\mathcal{N}_r}) \quad \forall r \in \mathcal{S}$$

es decir, un campo aleatorio es Markoviano si cada configuración tiene probabilidad positiva de ocurrir, y cada variable aleatoria es independiente de todas las variables aleatorias fuera de su vecindad, dada la realización de las variables aleatorias dentro de su vecindad.

Una vez definido el modelo para una imagen, estamos listos para plantear formalmente problemas de procesamiento de imágenes y al haber adoptado el modelo de campo aleatorio, lo primero que se nos podría ocurrir es definir funciones de distribución para F de tal manera que a las configuraciones $f \in \mathbb{F}$ se les asocie una mayor probabilidad si sus características satisfacen los requerimientos de una *solución* del problema planteado. De esta manera *resolver el problema* es equivalente a encontrar máximos de P . En capítulos posteriores se discutirán ésta y otras ideas, mientras tanto se ve que necesitaremos alguna manera de definir funciones de distribución para campos aleatorios.

2.3. Campos aleatorios de Gibbs

Definición: un campo aleatorio F es un *campo aleatorio de Gibbs* si para toda configuración $f \in \mathbb{F}$,

$$P(f) = \frac{1}{Z} e^{-U(f)}$$

donde la *función de potencial* U es de la forma

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f)$$

Aquí, V_c es una función que solamente depende de los valores de f en el clique c . Además Z es simplemente una constante de normalización.

El **Teorema de Hamersley-Clifford** establece que *un campo aleatorio F sobre un conjunto de sitios \mathcal{S} es Markoviano respecto a un sistema de vecindades \mathcal{N} si y sólo si F es un campo aleatorio de Gibbs sobre el conjunto de sitios \mathcal{S} , respecto al mismo sistema de vecindades.*

Lo anterior nos dice que si modelamos a \mathcal{S} como un Campo Aleatorio Markoviano con sistema de vecindades \mathcal{N} y necesitamos definir una función de densidad sobre el conjunto de configuraciones (imágenes) entonces basta definir las funciones **locales** V_C sobre cada clique.

2.4. Estimación Bayesiana

Consideremos el siguiente problema de clasificación. Supongamos que contamos con una observación $x \in \mathbb{R}^n$, la cual podemos asignar a una de M clases del conjunto $\mathcal{L} = \{q_1, q_2, \dots, q_M\}$. Suponiendo que existe un modelo probabilístico subyacente, nos gustaría elegir $y \in \mathcal{L}$ de tal forma que la probabilidad

$$P(Y = y \mid X = x)$$

sea máxima.

El modelo probabilístico antes mencionado define la probabilidad de observar x dado que proviene de la clase y . Esta probabilidad se llama *verosimilitud* y se expresa como:

$$P(x \mid y) := P(X = x \mid Y = y)$$

de este modo podemos usar la regla de Bayes para obtener

$$P(Y = y \mid X = x) = \frac{P(x \mid y)P(y)}{P(x)}$$

y dado que estamos buscando el valor de y que maximiza esta expresión, podemos ignorar el término $P(x)$. La distribución $P(y)$ es conocida como *información a priori*.

Un ejemplo conocido de este método se basa en suponer que la observación x proviene de una de M distribuciones normales con medias $\mu = \{\mu_1, \mu_2, \dots, \mu_M\}$ y varianzas $\Sigma = \{\Sigma_1, \Sigma_2, \dots, \Sigma_M\}$ lo cual podemos expresar como

$$[X \mid (Y = y)] \sim N(\mu_y, \Sigma_y)$$

entonces la verosimilitud queda completamente definida por el modelo

$$P(X = x \mid Y = y) = \phi(x; \mu_y, \Sigma_y)$$

donde $\phi(\cdot; \mu, \Sigma)$ es la función de densidad normal con media μ y matriz de covarianza Σ .

El problema general de estimación Bayesiana contempla la definición de una función $C : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}^+$ que asigna un costo no negativo a asignar a x a la clase \hat{y} cuando en realidad pertenece a la clase y . Esta función de costo en principio parece inútil ya que si conociéramos la clase a la cual pertenece x entonces nuestro problema de clasificación está resuelto, por tanto en realidad no podemos evaluar explícitamente

esta función. Lo que sí podemos hacer es evaluar el *valor esperado* del costo para una estimación \hat{y} respecto a la distribución *a posteriori* $P(y|x)$.

$$Q(\hat{y}) = E[C(\hat{y}, y) | \hat{y}] = \int C(\hat{y}, y) dP(y | x)$$

el estimador Bayesiano óptimo se define entonces como

$$\hat{y}^* = \operatorname{argmin}_y Q(y)$$

Es fácil determinar el estimador Bayesiano óptimo para diversas elecciones de C , en seguida muestro algunos ejemplos:

Ejemplo 1: sea $C(\hat{y}, y) = 1 - \delta(\hat{y} - y)$. Esta función asigna un costo unitario si el estimador es incorrecto y cero si es correcto. Entonces

$$Q(\hat{y}) = \int_{-\infty}^{\infty} C(\hat{y}, y) P(y | x) dy = \int_{-\infty}^{\infty} (1 - \delta(\hat{y} - y)) P(y | x) dy = 1 - P(\hat{y} | x)$$

el mínimo de esta función se alcanza cuando $P(\hat{y} | x)$ es máximo. Este estimador se conoce como *estimador MAP* (Maximizador de la distribución A Posteriori).

Ejemplo 2: sea $C(\hat{y}, y) = (\hat{y} - y)^2$, entonces

$$\frac{\partial Q}{\partial \hat{y}}(\hat{y}) = 2 \int_{-\infty}^{\infty} (\hat{y} - y) P(y | x) dy$$

igualando a cero obtenemos

$$\int_{-\infty}^{\infty} \hat{y} P(y | x) dy = \int_{-\infty}^{\infty} y P(y | x) dy \Rightarrow \hat{y} = \int_{-\infty}^{\infty} y P(y | x) dy = E_{P(y|x)} y$$

es decir, es estimador Bayesiano óptimo para esta función de costo es el promedio.

Ejemplo 3: sea $C(\hat{y}, y) = |\hat{y} - y|$, entonces

$$\begin{aligned} Q(\hat{y}) &= \int_{-\infty}^{\hat{y}} (\hat{y} - y) P(y | x) dy + \int_{\hat{y}}^{\infty} (y - \hat{y}) P(y | x) dy = \\ &\hat{y} \int_{-\infty}^{\hat{y}} P(y | x) dy - \int_{-\infty}^{\hat{y}} y P(y | x) dy + \int_{\hat{y}}^{\infty} y P(y | x) dy - \hat{y} \int_{\hat{y}}^{\infty} P(y | x) dy \end{aligned}$$

aplicando la regla de la cadena tenemos

$$\frac{\partial Q}{\partial \hat{y}} = \int_{-\infty}^{\hat{y}} P(y | x) dy + \hat{y} P(\hat{y} | x) - \hat{y} P(\hat{y} | x) - \hat{y} P(\hat{y} | x) + \hat{y} P(\hat{y} | x) - \int_{\hat{y}}^{\infty} P(y | x) dy$$

igualando a cero llegamos a

$$\int_{-\infty}^{\hat{y}} P(y | x) dy = \int_{\hat{y}}^{\infty} P(y | x) dy$$

esto es, el estimador Bayesiano óptimo es la mediana respecto a la distribución a posteriori.

Lo anterior muestra la dinámica a seguir para encontrar estimadores Bayesianos óptimos, la cual resumimos en los siguientes 5 pasos:

- Diseñar un modelo de observación que defina la verosimilitud $P(x | y)$.
- Fijar información conocida acerca de la distribución de y , definiendo la densidad $P(Y = y)$.
- Usar la verosimilitud y la distribución *a priori* para expresar la distribución *a posteriori* $P(Y = y | X = x)$.
- Elegir una función de costo $C(\hat{y}, y)$.
- Elegir como estimador \hat{y} , el que minimiza el valor esperado de la función de costo $Q(\hat{y})$.

Capítulo 3

Problemas clásicos de procesamiento de imágenes de bajo nivel

*Science never solves a problem
without creating ten more.*

George Bernard Shaw.

En este capítulo describo a grandes rasgos algunos de los problemas clásicos de procesamiento de imágenes. El objetivo es mostrar ejemplos de los pasos a seguir en la modelación de estos problemas hasta traducir nuestro problema original en un problema de optimización.

3.1. Restauración

En el proceso de captura, siempre están presentes fenómenos que introducen ruido a las imágenes. Debido a que este problema se presenta en todo tipo de imágenes, el problema de restauración ha sido objeto de un gran trabajo de investigación.

Para hacer uso de la modelación mediante campos aleatorios Markovianos y la estimación Bayesiana, típicamente modelamos el ruido en las imágenes como un conjunto de variables aleatorias independientes e idénticamente distribuidas, de tal forma que si g es la imagen observada y f es la imagen “verdadera”, entonces

$$g_r = f_r + \eta_r$$

si suponemos

$$\eta_r \sim N(0, \sigma^2)$$

entonces la *verosimilitud* para uno de los sitios es

$$v(r) = P(g_r | f_r) = \phi(g_r - f_r; \mu, \sigma^2)$$

y como asumimos que las variables aleatorias son i.i.d, la verosimilitud *conjunta* es

$$v = P(g | f) = \prod_{r \in \mathcal{S}} \phi(g_r - f_r; \mu, \sigma^2)$$

El siguiente paso es fijar información a priori acerca de f . Modelaremos f como un campo aleatorio Markoviano. Lo que *conocemos* acerca de f es que se trata de una función *suave*. Recordando la propiedad de los campos Gibbsianos, podemos definir la probabilidad $P(f)$ simplemente definiendo funciones de potencial *locales*. Para decir que f es suave, decimos que f_r es muy parecido a f_s si r, s son vecinos entre si. Consideremos el caso de la 4-vecindad. Para un cliqué $C = \langle r, s \rangle$, definiremos los potenciales locales V_C de forma que *penalicen* las diferencias entre pixeles vecinos, esto es:

$$V_C(f) = \lambda(f_r - f_s)^2$$

donde λ es un parámetro libre que se utiliza para controlar el efecto de la distribución a priori sobre la distribución a posteriori. Idealmente este parámetro se ajusta manualmente hasta obtener resultados convenientes para un conjunto de imágenes representativas de la aplicación de interés para mantenerlo fijo posteriormente. En particular, para esta función de potencial, valores grandes de λ se traducen en imágenes cada vez mas suaves como resultado de la restauración.

Elijamos la función de costo como $C(\hat{f}, f) = 1 - \delta(\hat{f} - f)$, ya vimos antes que el estimador Bayesiano óptimo es aquel que maximiza la distribución a posteriori.

$$P(f | g) = \left[\prod_{r \in \mathcal{S}} \phi(g_r - f_r; \mu, \sigma^2) \right] \exp\left(- \sum_{C \in \mathcal{C}_2} V_C(f)\right)$$

En esta clase de problemas, es conveniente tomar el logaritmo. Así, para maximizar la distribución a posteriori, basta minimizar el negativo de su logaritmo:

$$-\log P(f | g) = \frac{1}{2\sigma^2} \sum_{r \in \mathcal{S}} (f_r - g_r)^2 + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} (f_r - f_s)^2$$

El primer término es conocido en la literatura como *término de datos* y es el término que obliga a que $f_r \approx g_r$, es decir, que la restauración f se parezca a la imagen original g . El segundo término se conoce como *término de regularización* y es el que obliga a que $f_r \approx f_s$ donde r, s son píxeles vecinos, es decir, que la restauración sea suave. La restauración final depende del comportamiento del término de regularización respecto al término de datos.

El último paso en el procedimiento de estimación Bayesiana consiste en encontrar el mínimo de esta función, esto se puede hacer usando cualquier método de optimización basado en descenso de gradiente[30].

En la figura 3.6-b se ilustra el estimador Bayesiano óptimo obtenido usando este potencial. Se observa el conocido fenómeno de “sobre suavizamiento”. Esto se debe a que solamente introdujimos como información a priori, el que el estimador óptimo fuera suave y para esto elegimos como término de regularización la norma L_2^2 que obliga a que los píxeles vecinos sean parecidos sin importar la magnitud de su diferencia, pero en lugar de esto queremos que el potencial sea capaz de diferenciar las zonas de la imagen que deben ser suavizadas de las que no deben suavizarse. La ventaja de esta modelación es que ahora es sencillo modificar este detalle con sólo modificar la función V_C . De este modo surgen los *potenciales robustos* que se muestran en la siguiente sección.

3.1.1. Potenciales robustos

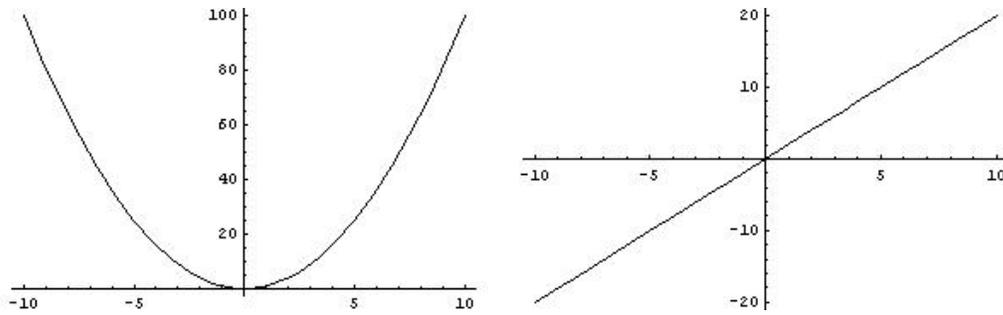
Para evitar que se pierdan los bordes al restaurar una imagen, necesitamos cambiar la información a priori. En lugar de pedir suavidad global, pediremos *suavidad a pedazos*, la forma más simple de hacer esto es notar que un *borde* está definido por dos píxeles vecinos cuya diferencia es grande respecto a las diferencias debidas al ruido. Para modelar esto utilizamos los potenciales robustos, cada uno tiene diferentes características y la elección de alguno depende de la aplicación.

$$V_C(f) = \lambda \rho(f_r - f_s)$$

Los siguientes son ejemplos de potenciales utilizados en el problema de restauración. Muestro para cada uno una gráfica del potencial y su derivada.

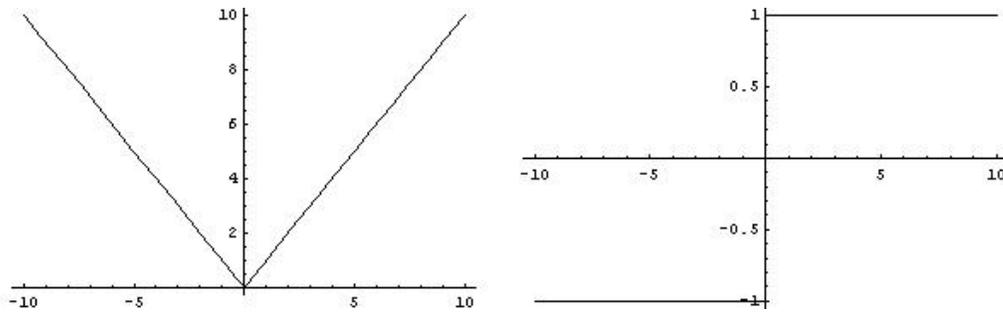
- Norma L_2^2

$$\rho(x) = x^2$$

Figura 3.1: Norma L_2^2 y su derivada.

- Norma L_1

$$\rho(x) = |x|$$

Figura 3.2: Norma L_1 y su derivada.

- Minimax de Huber

$$\rho(x) = \begin{cases} \frac{x^2}{2\epsilon} + \frac{\epsilon}{2} & \text{si } |x| < \epsilon \\ |x| & \text{si no} \end{cases}$$

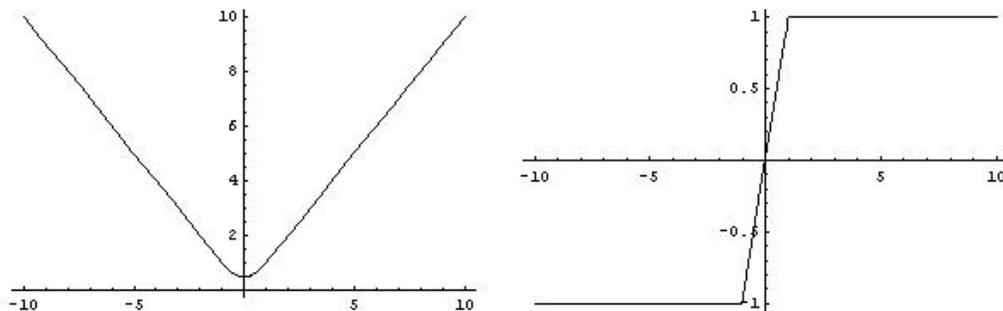


Figura 3.3: Minimax de Huber y su derivada.

- Hampel et al., 1986

$$\rho(x) = \begin{cases} x^2 & \text{si } |x| < \sqrt{\beta} \\ \beta & \text{si no} \end{cases}$$

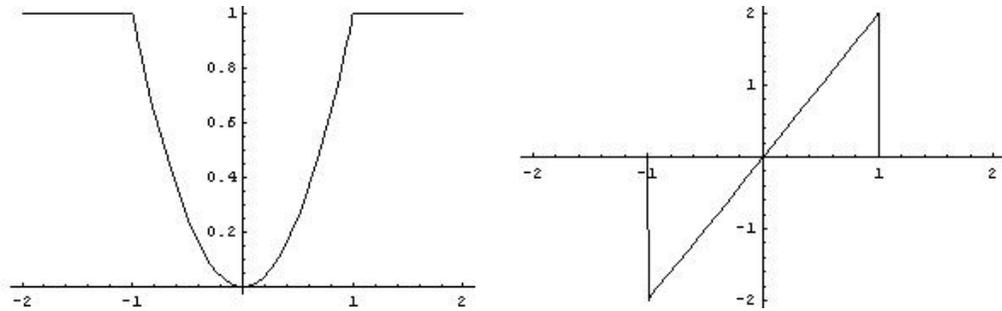


Figura 3.4: Potencial de Hampel y su derivada.

- Geman - Mc.Clure 1987

$$\rho(x) = \frac{x^2}{c + x^2}$$

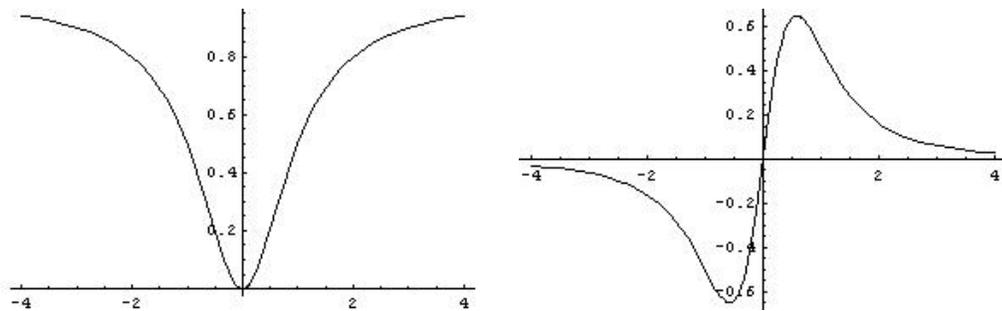


Figura 3.5: Potencial de Geman-Mc.Clure y su derivada.

En la figura 3.6 se muestran los resultados de la restauración de la conocida imagen de Lenna (normalizada al rango dinámico $[0, 1]$) contaminada con ruido Gaussiano con $\sigma = 0.05$ usando diferentes potenciales robustos. Recordemos que el potencial, al ser aplicado a nuestro problema de restauración, será evaluado en $x = (f_r - f_s)$ donde r, s son pixeles vecinos, es decir, se aplicará $\rho(f_r - f_s)$, por lo que la forma de analizar el comportamiento de los potenciales robustos usando las gráficas consiste en considerar x como “la diferencia entre pixeles vecinos”. Una característica general de los potenciales robustos es que son funciones simétricas, es decir $\rho(x) = \rho(-x)$ así que basta observar el comportamiento para $x \geq 0$. Otro punto de vista que puede ser útil para comprender el efecto de los potenciales, es considerar la derivada como una medida de la “fuerza” con que se penalizan las diferencias grandes y pequeñas entre pixeles vecinos, así que la derivada define la forma como aumenta o disminuye la fuerza con que actúa el potencial al aumentar o disminuir la diferencia.

En la imagen 3.6 b), como se mencionó antes, podemos apreciar el efecto de la norma L_2^2 , los bordes se suavizan debido a que la norma L_2^2 penaliza las diferencias entre pixeles vecinos cada vez con más

fuerza (a mayor discrepancia mayor penalización), así que los bordes (diferencias grandes), tenderán a ser eliminados. Por otro lado podemos observar que los bordes se respetan bastante bien al utilizar la norma L_1 , esto se debe a que la penalización es la misma para diferencias pequeñas que para diferencias grandes entre pixeles vecinos. Si consideramos un par de pixeles vecinos r, s en un borde, la diferencia $|f_r - f_s|$ será relativamente grande, el término de datos obliga $f_r \approx g_r$ y $f_s \approx g_s$ con una fuerza relativamente grande mientras que el término de regularización (usando la norma L_1) presiona a que $f_r \approx f_s$ pero con una fuerza menor que la del término de datos, así que el borde se respeta. El otro efecto de la norma L_1 y que también se debe a que las diferencias se penalizan de manera constante es que las regiones de diferencias pequeñas se penalizan con más fuerza que al usar la norma L_2^2 , este efecto **no** es deseable para superficies cuyo brillo varía suavemente a lo largo de la imagen ya que estas diferencias serán penalizadas y al final obtendremos un efecto de *superficie escalonada* en lugar de una superficie suave (este efecto se aprecia por ejemplo en la frente y nariz de Lenna de la figura 3.6 c)). Este último fenómeno se resuelve de manera muy natural al usar el potencial de Huber que es una combinación de las normas L_1 y L_2^2 . Se utiliza la norma L_1 para valores grandes de $|x|$ para aprovechar sus ventajas al preservar los bordes y la norma L_2^2 para valores pequeños para evitar el efecto de escalonado (nótese la diferencia en la nariz y la frente de Lenna en la figura 3.6 d) mientras que al mismo tiempo se respetan los bordes del sombrero y otras regiones).

Siguiendo estas ideas se han desarrollado otros potenciales robustos como por ejemplo el potencial de Hampel. La idea de este potencial es fijar un umbral ($\sqrt{\beta}$) a partir del cual se considera que una diferencia es un borde. Si existe un borde entre un par de pixeles vecinos, no hay razón para penalizar la diferencia entre ellos así que la derivada es cero más allá de este umbral, por otro lado si no existe un borde entre un par de pixeles vecinos entonces se utiliza la norma L_2^2 . La idea de este potencial es bastante simple pero presenta diversos problemas, entre ellos se encuentra el que el potencial no es diferenciable en $\pm\sqrt{\beta}$ por lo que hay que tener cuidado de considerar especialmente estos casos. El siguiente problema es que el criterio de lo que es o no un borde se basa en el umbral $\sqrt{\beta}$, así que solamente se respetarán los bordes muy cercanos a este umbral, este aspecto es particularmente inconveniente en regiones en las que el borde entre dos objetos no está definido por un umbral constante, es decir, el borde se hace mas marcado o menos marcado a lo largo de los objetos que lo definen, en este caso el efecto es el de un borde discontinuo. Un ejemplo es el sombrero de Lenna, nótese que el borde con el fondo es muy marcado en una región e inexistente en otra. El primer problema se resuelve usando el potencial de Geman-Mc.Clure, el cual en lugar de definir un umbral rígido como el de Hampel, tiene un comportamiento decreciente

asintótico hacia cero al crecer la diferencia entre pixeles vecinos, con esto la diferencia no necesita llegar a un umbral dado para tener un efecto. El fenómeno de la discontinuidad de los bordes está presente al igual que con el potencial de Hampel (Figura 3.6 e)).



Figura 3.6: Restauración usando diferentes potenciales(ver texto).

La propiedad de “suavidad” (*smooth prior*) es ampliamente utilizada como conocimiento a priori, no solamente en el problema de restauración. Este criterio se utiliza para “propagar” la información hacia regiones en las que la *verosimilitud* no ayuda mucho, como veremos en los siguientes ejemplos. Naturalmente el estudio de potenciales robustos y sus propiedades también aplica a estos otros problemas.

3.2. Estimación de la disparidad estereoscópica

La forma como percibimos la distribución espacial de los objetos en el entorno es una capacidad que tenemos y en la que rara vez pensamos, es muy natural ver un objeto y darnos una idea de su tamaño y de la distancia a la que se encuentra. Un experimento interesante es cubrir completamente uno de nuestros ojos y viendo solo con uno de ellos tratar de sujetar un objeto pequeño mas o menos lejano. El fenómeno que se observa es que es ligeramente más difícil no “fallar” en el primer intento.

La capacidad que tenemos de percibir esta distribución espacial se logra gracias a que en todo momento contamos con dos imágenes de la misma escena vista desde posiciones ligeramente diferentes, una por cada ojo. El cerebro combina estas dos imágenes y de este modo “sabe” la distancia a la que se encuentran los objetos.

Una forma de modelar este “sistema óptico” se muestra en la figura 3.7. Modelamos el par de imágenes mediante un par de *planos de proyección* sobre los cuales se proyecta la escena. Cada proyección se realiza respecto a un centro de proyección. Los puntos P, C_1, C_2 se encuentran en un plano llamado *plano epipolar*. La intersección de este plano con los planos de proyección definen las *líneas epipolares*. La observación importante es que todos los puntos en la línea $\overline{P, C_1}$ se proyectan sobre el mismo punto Q_1 en el plano de proyección Σ_1 , pero serán proyectados sobre la línea epipolar $\overline{Q_2, e_2}$ en el plano de proyección Σ_2 , y viceversa [1].

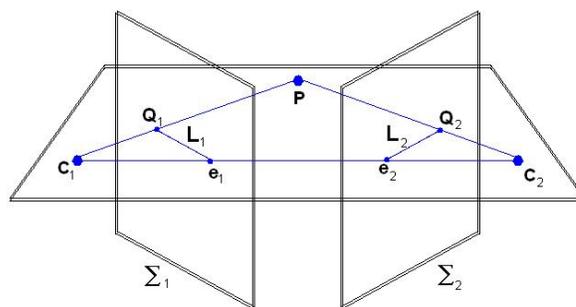


Figura 3.7: Modelo de visión estereo.

Debido a la gran cantidad de aplicaciones que surgen de este problema, se ha desarrollado, y se sigue desarrollando mucho trabajo de investigación en esta dirección. Sólo por mostrar un ejemplo, para el caso particular en que las líneas epipolares son horizontales y coplanares, nuestro problema se reduce a

encontrar, para cada pixel de la imagen izquierda r , un desplazamiento d_r horizontal de modo que

$$f_{izq}(r) = f_{der}(r + d_r)$$

siguiendo los pasos para estimación Bayesiana, podemos establecer un modelo de observación de la forma

$$f_{izq}(r) = f_{der}(r + d_r) + \eta_r$$

con $\eta_r \sim N(0, \sigma^2)$. Esto define la verosimilitud, el siguiente paso es fijar la información a priori, para lo cual podemos utilizar las ideas antes mencionadas acerca de la condición de suavidad (smooth prior) para definir, por ejemplo, las funciones de potencial

$$V_{\langle r,s \rangle}(d) = \|d_r - d_s\|^2$$

utilizar potenciales robustos, etc. A pesar de lo burdo de este modelo, encierra algunas ideas básicas utilizadas en todos los métodos de estimación de la disparidad y estimación, sin embargo, no es un problema fácil y los métodos más actuales se basan en ideas muy diversas y sofisticadas[1][2][14].

3.3. Estimación de flujo óptico

El problema de la estimación del flujo óptico surge del problema de la estimación de movimiento en una secuencia de video.

El problema de reconocer el movimiento de objetos vistos en una secuencia de imágenes se ha tratado de resolver usando muchos modelos que tratan de describir las características generales de dicho movimiento. El primer problema con el que uno se encuentra es el de definir de manera general qué es lo que entendemos por movimiento. De manera formal podríamos tratar de definir el movimiento de una partícula en un espacio E en el tiempo t_0 , como la velocidad de dicha partícula en el tiempo t_0 , de esta forma, una partícula que sigue una trayectoria $c : T \rightarrow E$ tendría una velocidad igual a $v(t_0) = \frac{\partial c}{\partial t}(t_0)$. Entonces el “campo de movimiento” de E en el tiempo t_0 , queda definido por una función $F(x) = \frac{\partial c_x}{\partial t}(t_0)$ donde $c_x(t)$ es la trayectoria que sigue la partícula que se encuentra en la posición x en el tiempo t_0 . Dadas dos imágenes de la misma escena tomadas en dos instantes t_0 y t_1 , podríamos describir aproximadamente la proyección del desplazamiento de cada partícula en el intervalo de tiempo $[t_0, t_1]$, si tuviéramos una forma de reconocer cada partícula de la escena en cada una de las dos imágenes. Hay que notar que sólo vamos a conocer la proyección del campo de movimiento sobre el

mismo plano donde se proyectan los objetos para ser representados en las imágenes.

Es aquí donde surge el problema de encontrar los rasgos que describen a las partículas para poder encontrarlas en diferentes etapas de su movimiento. Como es de esperarse un conjunto de partículas pueden tener muchos rasgos tan parecidos que es difícil diferenciarlas, por lo que el resultado que obtengamos al resolver el problema del campo de movimiento va a depender de las características que elijamos para comparar a las partículas. Como estamos trabajando con imágenes, lo primero que a uno se le ocurre es usar el brillo de los píxeles (nivel de gris) como rasgo característico; esperamos que una misma partícula conserve su brillo a lo largo de su movimiento. De esta manera, la estimación de la proyección del campo de movimiento (que desde ahora llamaré simplemente *velocidad*) se traduce en encontrar el campo vectorial que describe la forma como se mueven los tonos de gris en la imagen, esto es, el *flujo óptico*. En la figura 3.8 se muestran dos cuadros de la famosa secuencia del taxi de Hamburgo, el taxi en el video gira y se aleja de la cámara, por tanto el movimiento es en tres dimensiones pero solamente podemos determinar la proyección del movimiento sobre el plano de proyección de la cámara, sin embargo, el flujo óptico está bien definido y es aquel que describe el movimiento de los tonos de gris a lo largo de la región de vista.



Figura 3.8: Dos frames de la secuencia del “Taxi de Hamburgo”.

A diferencia del problema de la estimación de la disparidad, en la estimación del flujo óptico no tenemos el concepto de “líneas epipolares” sobre las cuales restringir la búsqueda. En este caso los modelos que se utilizan son desplazamientos en \mathbb{R}^2 y nuevamente, las ideas desarrolladas para el problema de restauración y de los potenciales robustos también aplican a este problema. Los trabajos de Michael J. Black han sido de los más innovadores en esta area [4]. Otras técnicas basadas en información de fase han tenido también mucho éxito [15].

3.4. Segmentación

El problema de segmentación puede definirse en pocas palabras como encontrar una partición de una imagen observada g de forma que los elementos de cada región compartan algún *rasgo* en común. Estas características son modeladas a partir de funciones paramétricas ϕ_{θ_i} , es decir, si $\{R_i\}_{i=1}^M$ es la partición que estamos buscando entonces ϕ_{θ_k} es un *buen* modelo para $g|_{R_k}$ (g restringida a R_k).

La formulación clásica del problema de segmentación parte del supuesto de que la imagen observada es una versión distorcionada de una imagen original que fue obtenida a partir de una partición que es justamente la que queremos obtener. En la figura 4.4 se ilustra este modelo de generación - observación. En el primer paso, un campo de etiquetas f , el cual modelamos como un Campo Aleatorio Markoviano es generado con probabilidad

$$P(f) = \frac{1}{Z} \exp(-U(f))$$

donde

$$U(f) = \sum_{C \in \mathcal{C}} V_C(f)$$

para definir el potencial V_C para los cliques $C \in \mathcal{C}$ generalmente se utiliza el potencial de Ising eligiendo el sistema de 4-vecindades. Este potencial, para un cliqué $C = \langle r, s \rangle$ está dado por

$$V_{\langle r, s \rangle}(f) = \begin{cases} \lambda & \text{si } f(r) \neq f(s) \\ 0 & \text{si } f(r) = f(s) \end{cases}$$

donde λ , al igual que se utilizó en el problema de restauración, es un parámetro que se utiliza para controlar el efecto de este potencial. El potencial de Ising es la contraparte discreta de la condición de suavidad (smooth prior) que se discutió en el problema de restauración, en este caso cada sitio de f solo toma valores en un conjunto discreto \mathcal{L} (en este caso las diferentes etiquetas de la segmentación).

En el siguiente paso, para cada sitio $r \in \mathcal{S}$ evaluamos $\tilde{f}(r) = \phi_{\theta_k}(r)$ tal que $f(r) = q_k$ (la etiqueta asociada al sitio r es q_k). Finalmente, la imagen original \tilde{f} es perturbada con ruido aditivo, el cual es modelado mediante las variables aleatorias independientes e idénticamente distribuidas η_r

$$g(r) = \tilde{f}(r) + \eta_r$$

Entonces el modelo de observación puede expresarse como

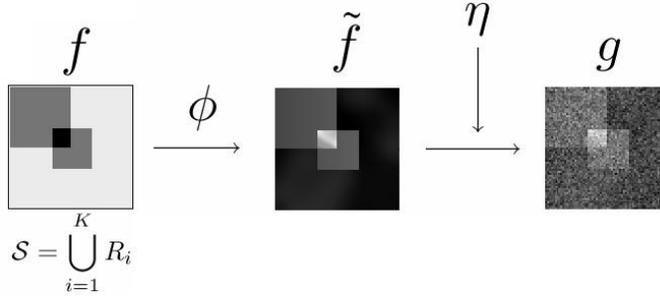


Figura 3.9: Modelo de generación - Observación.

$$g(r) = \sum_{i=1}^M b_i(r) \phi_{\theta_i}(r) + \eta_r$$

donde

$$b_i(r) = \begin{cases} 1 & \text{si } f(r) = q_i \\ 0 & \text{si } f(r) \neq q_i \end{cases} \quad (3.1)$$

El problema de segmentación se traduce en encontrar valores “óptimos” para $b_i(r)$ y θ_i , los parámetros de los modelos. Como criterio de optimalidad usamos la distribución a-posteriori: buscamos los valores de $b_i(r)$ y θ_i que maximicen dicha distribución

$$P(b, \theta | g) = \frac{1}{P(g)} P(g | b, \theta) P(b) P(\theta) = \prod_{r \in S} \left[\sum_{k=1}^M b_k(r) v_k(r) \right] \exp \left(- \sum_{C \in \mathcal{C}} V_C(b) \right) P(\theta)$$

El factor $P(\theta)$ puede asumirse constante (distribución uniforme para los parámetros). También es común en algunas aplicaciones modelar θ como un campo aleatorio markoviano y definir su propio potencial como información a priori

$$P(\theta) = \frac{1}{Z} \exp(-W(\theta))$$

Maximizar esta distribución es equivalente a minimizar la función de energía (tomando logaritmo)

$$U(b, \theta) = \sum_{r \in S} \sum_{i=1}^M -b_i(r) \log v_i(r) + \sum_{C \in \mathcal{C}} V_C(b) + W(\theta)$$

Lo que hace complejo este problema de optimización es que las variables $b_i(r)$ solo pueden tomar uno de los valores $\{0, 1\}$, es decir, se trata de un problema de optimización combinatoria y para resolverlo

aproximadamente es necesario utilizar métodos como el método de *modas condicionales iteradas* [3] (ICM por *Iterated Conditional Modes*).

Dada una configuración inicial f^0 de f , ICM iterativamente actualiza cada variable $f_r^{(t)}$ como $f_r^{(t+1)}$ tal que maximiza la probabilidad condicional a posteriori

$$P(f_r^{(t+1)} | g, f_{\mathcal{N}_r})$$

donde $f_{\mathcal{N}_r}$ es el conjunto de variables en la vecindad de r . Este proceso de actualización se repite hasta convergencia. Se invita al lector interesado en el algoritmo ICM a revisar los detalles en las referencias [3]. Otros algoritmos de optimización combinatoria pueden consultarse en una excelente comparación hecha por Stan Z. Li. [26]

Lo anterior da como resultado los estimadores MAP $\hat{b}, \hat{\theta}$ para b y θ . Mas en general, en el marco de estimación bayesiana, podemos definir una *función de costo* $C(\hat{b}, \hat{\theta}, b, \theta)$. El valor esperado del costo respecto a la distribución a posteriori es

$$Q(\hat{b}, \hat{\theta}) = E[C(\hat{b}, \hat{\theta}, b, \theta) | g] = \sum_{b \in \mathcal{B}} \int_{\theta \in \Theta} C(\hat{b}, \hat{\theta}, b, \theta) P(b, \theta | g) d\theta$$

La optimización de esta función se realiza de manera iterativa optimizando alternadamente respecto a \hat{b} dejando fijo un estimador $\bar{\theta}$ para θ y respecto a $\hat{\theta}$ dejando fijo un estimador \bar{b} para b (ver algoritmo 1).

$$Q_b(\hat{b}; \bar{\theta}) = E[C(\hat{b}, \bar{\theta}, b, \theta) | g, \bar{\theta}] = \sum_{b \in \mathcal{B}} C(\hat{b}, \bar{\theta}, b, \theta) P(b, \bar{\theta} | g)$$

$$Q_\theta(\bar{b}, \hat{\theta}) = E[C(\bar{b}, \hat{\theta}, b, \theta) | g, \bar{b}] = \int_{\theta \in \Theta} C(\bar{b}, \hat{\theta}, b, \theta) P(\bar{b}, \theta | g) d\theta$$

Una clase importante de funciones de costo es de la forma

$$C_b(\hat{b}, b) = \sum_{r \in \mathcal{S}} C_r(\hat{b}(r), b(r))$$

en este caso

$$\begin{aligned} Q_b(\hat{b}; \bar{\theta}) &= \sum_{b \in \mathcal{B}} P(b, \bar{\theta} | g) \sum_{r \in \mathcal{S}} C_r(\hat{b}(r), b(r)) = \sum_{r \in \mathcal{S}} \sum_{b \in \mathcal{B}} P(b, \bar{\theta} | g) C_r(\hat{b}(r), b(r)) = \\ &= \sum_{r \in \mathcal{S}} \sum_{k=1}^M \sum_{b: b(r)=e_k} P(b, \bar{\theta} | g) C_r(\hat{b}(r), b(r)) = \sum_{r \in \mathcal{S}} \sum_{k=1}^M C_r(\hat{b}(r), e_k) \sum_{b: b(r)=e_k} P(b, \bar{\theta} | g) = \end{aligned}$$

Algorithm 1 Algoritmo SP

-
- 1: Elegir valores iniciales θ_0 ;
 - 2: Fijar la tolerancia $\epsilon > 0$;
 - 3: Inicializar $s = \epsilon, i = 0$;
 - 4: **while** $s \geq \epsilon$ **do**
 - 5: $i++$;
 - 6: Encontrar $b_i = \operatorname{argmin}_b \{Q_b(b, \theta_{i-1})\}$;
 - 7: Encontrar $\theta_i = \operatorname{argmin}_\theta \{Q_\theta(b_i, \theta)\}$;
 - 8: $s = \text{error}(b_i, \theta_i)$;
 - 9: **end while**
 - 10: Finalizar con solución b_i, θ_i ;
-

$$\sum_{r \in \mathcal{S}} \sum_{k=1}^M C_r(\hat{b}(r), e_k) \pi_k(r)$$

Si C_r son funciones no negativas, entonces minimizar esta expresión es equivalente a minimizar por separado cada uno de los sumandos. El estimador óptimo de $b(r)$ es entonces

$$\hat{b}(r)^* = \operatorname{argmin}_{b(r)} \sum_{k=1}^M C_r(b(r), e_k) \pi_k(r)$$

donde

$$\pi_k(r) = \sum_{b: b(r)=e_k} P(b, \bar{\theta} | g) = P(b(r) = e_k, \bar{\theta} | g)$$

define la distribución marginal (discreta) a posteriori $\pi(r)$ en el sitio r .

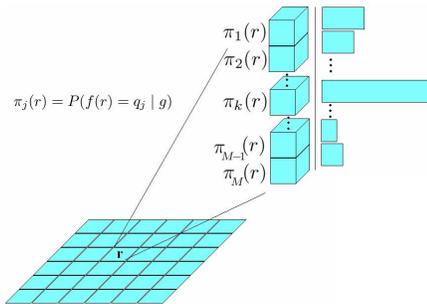


Figura 3.10: Distribuciones marginales a posteriori.

En particular, la función de costo que penaliza el número de sitios erróneos es

$$C_b(\hat{b}, b) = \sum_{r \in \mathcal{S}} [1 - \delta(\hat{b}(r) - b(r))]$$

y por tanto, el estimador óptimo \hat{b}^* para b está definido por

$$\hat{b}^*(r) = \operatorname{argmax}_{e_k} \pi_k(r)$$

Lo anterior muestra que es posible encontrar estimadores Bayesianos óptimos fácilmente para una clase de funciones de costo muy general, con “solo” encontrar las distribuciones marginales $\pi(r)$. El problema es que aunque tenemos una expresión explícita para $\pi_k(r)$, actualmente no es posible calcularlo en un tiempo razonable ya que es necesario considerar todos los posibles valores de b .

Capítulo 4

Modelos utilizados en el problema de segmentación

*Whenever anyone says, 'theoretically,'
they really mean, 'not really.'*

Dave Parnas.

El capítulo anterior finalizó con el planteamiento de un problema: calcular las probabilidades marginales a posteriori $\pi_k(r)$. En las siguientes secciones se describen algunas técnicas que han sido desarrolladas para encontrar *buenas* aproximaciones de estas marginales.

4.1. Cadenas de Markov - Monte Carlo (MCMC)

Antes de explicar la aplicación de MCMC al problema de segmentación de imágenes es necesario introducir algunos conceptos. Nos restringiremos al estudio de cadenas de Markov con espacio de estados finito.

Sea $R = \{q_1, q_2, \dots, q_M\}$ un conjunto finito y $X = \{X_1, X_2, \dots\}$ una sucesión de variables aleatorias que toman valores en R . Llamaremos a R el *espacio de estados* y a sus elementos, *estados*.

Definición: decimos que $X = \{X_1, X_2, \dots\}$ es una *cadena de Markov de orden k* si para cualquier $n > k$ se cumple que

$$P(X_n | X_1, X_2, \dots, X_{n-1}) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-k})$$

Definición: supongamos que X es una cadena de Markov de orden 1, definimos la *matriz de transición* de X como la matriz E definida por

$$E_{i,j} = P(X_t = q_j \mid X_{t-1} = q_i)$$

Dado que cada variable aleatoria toma valores en R , cada fila de E representa una función de densidad discreta, y por tanto para cada $1 \leq i \leq M$, se cumple

$$\begin{aligned} E_{i,j} &\geq 0 \\ \sum_{j=1}^M E_{i,j} &= 1 \end{aligned} \tag{4.1}$$

Una matriz que cumple (4.1) se llama *matriz estocástica*.

Una de las preguntas que nos podemos hacer acerca de una cadena de Markov es cuál es la probabilidad de que en el tiempo t , el estado de la cadena sea algún $q_j \in R$. Responderemos esta pregunta con las siguientes consideraciones: sea X una cadena de Markov de orden 1 con matriz de transición M . Denotemos por α_t al vector cuya entrada j es $\alpha_t(j) = P(X_t = q_j)$, entonces por la ley de la probabilidad total

$$\alpha_{t+1}(j) = P(X_{t+1} = q_j) = \sum_{i=1}^M P(X_t = q_i)P(X_{t+1} = q_j \mid X_t = q_i) = \sum_{i=1}^M \alpha_t(i)E_{i,j}$$

o en forma matricial

$$\alpha_{t+1} = E^T \alpha_t$$

Por tanto, si conocemos α_0 , podemos calcular la probabilidad de que la cadena se encuentre en cualquier estado en cualquier tiempo:

$$\alpha_t = (E^T)^t \alpha_0$$

Definición: dada una cadena de Markov X con espacio de estados R , decimos que un estado $b \in R$ es *accesible* desde $a \in R$ si existe una sucesión de estados $\{x_1, x_2, \dots, x_k\}$, $x_i \in R$ tal que

$$a = x_1$$

$$b = x_k$$

$$P(X_t = x_t \mid X_{t-1} = x_{t-1}) > 0 \quad \forall \quad 1 < t \leq k$$

Definición: decimos que dos estados $a, b \in R$ son *mutuamente accesibles* si a es accesible desde b y b es accesible desde a .

Definición: una cadena de Markov X con valores en R es *regular* (o *irreducible*) si cualesquiera dos estados de R son mutuamente accesibles. A la matriz de transición de una cadena de Markov regular también le llamamos *regular*.

Definición: decimos que un estado $q_j \in R$ tiene *periodo* k si solo es posible regresar a dicho estado después de un múltiplo de k pasos. Si un q_j tiene periodo 1 decimos que es un estado *no periódico*. Si el periodo de q_j es mayor que 1, decimos que q_j es un estado *periódico*.

Definición: decimos que una cadena de Markov regular es *no periódica* si sus estados son no periódicos.

Una cuestión de particular interés es la existencia del límite

$$\lim_{t \rightarrow \infty} \alpha_t = \alpha^*$$

Al vector α^* se le llama *distribución estacionaria*. Esto significa que para t suficientemente grande la probabilidad de que X_t tome el valor q_j es muy próxima a $\alpha^*(j)$.

El teorema fundamental de las cadenas regulares de Markov¹ nos dice cuándo una cadena de Markov tiene una distribución estacionaria.

Teorema: si E es una matriz de transición regular no periódica entonces

1. Las potencias E^n se aproximan a una matriz de probabilidad límite A .
2. Cada fila de A es el mismo vector de probabilidad $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$.
3. Los componentes de α son positivos.

Los métodos MCMC están basados en el siguiente teorema

Teorema (Teorema Ergódico): si $\{X_1, X_2, \dots\}$ es una cadena de Markov regular no periódica entonces, con probabilidad 1, para cualquier función acotada $f : R \rightarrow \mathbb{R}$ se cumple que

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n f(X_t) = \sum_{q_j \in R} \alpha(j) f(q_j)$$

¹Una prueba de este teorema se encuentra en el libro de Kemeny y Snell[22].

donde α es la distribución estacionaria[31].

Notemos que este resultado es útil cuando el espacio de estados R es muy grande, de tal forma que sea más conveniente construir una cadena de Markov suficientemente grande (aunque mucho menos grande que la cardinalidad de R) que evaluar f en todo el espacio de estados. Este es precisamente nuestro caso, R es el conjunto de todas las posibles imágenes (etiquetados) que podemos dar como solución a nuestro problema de segmentación.

Sea \mathcal{F} el conjunto de todas las posibles segmentaciones con etiquetas $\mathcal{L} = \{q_1, q_2, \dots, q_M\}$. Supongamos que de alguna manera podemos construir una cadena de Markov $\{f^{(1)}, f^{(2)}, \dots\}$ regular, no periódica, con espacio de estados \mathcal{F} y con la distribución a posteriori como distribución estacionaria $\alpha(f) = P(f | g)$. Denotemos por $f_r^{(j)}$ el valor de la imagen $f^{(j)}$ en el sitio $r \in \mathcal{S}$. Definamos la función $\delta_j(\cdot; r) : \mathcal{F} \rightarrow \mathbb{R}$ por

$$\delta_j(f; r) \begin{cases} 1, & \text{si } f_r = q_j \\ 0, & \text{si } f_r \neq q_j \end{cases}$$

al ser una función acotada, el teorema ergódico nos dice que

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n \delta_j(f^{(t)}; r) = \sum_{f \in \mathcal{F}} \alpha(f) \delta_j(f; r) = \sum_{f: f_r = q_j} \alpha(f) = \pi_j(r)$$

esta es justamente la probabilidad marginal a posteriori definida en el capítulo anterior. La primera expresión es simplemente el número de veces que el sitio r tomó el valor q_j , dividido por n , es decir, podemos aproximar las distribuciones marginales a posteriori, usando las distribuciones empíricas.

La figura 4.1 ilustra el procedimiento antes descrito. Aquí suponemos que la sucesión $\{f^{(1)}, f^{(2)}, \dots\}$ es una cadena de Markov con las propiedades antes descritas. Para cada una de las muestras $f^{(t)}$ definimos las variables binarias $b_k^{(t)}(r)$, $1 \leq M$, $r \in \mathcal{S}$ por

$$b_k^{(t)}(r) = \begin{cases} 1 & \text{si } f_r^{(t)} = q_k \\ 0 & \text{si } f_r^{(t)} \neq q_k \end{cases}$$

como se ilustra en esta figura, al promediar estos vectores binarios componente a componente, obtenemos justamente las probabilidades marginales empíricas $p_k(r)$, las cuales son estimadores insesgados de las verdaderas marginales $\pi_k(r)$, es decir $E[p] = \pi$.

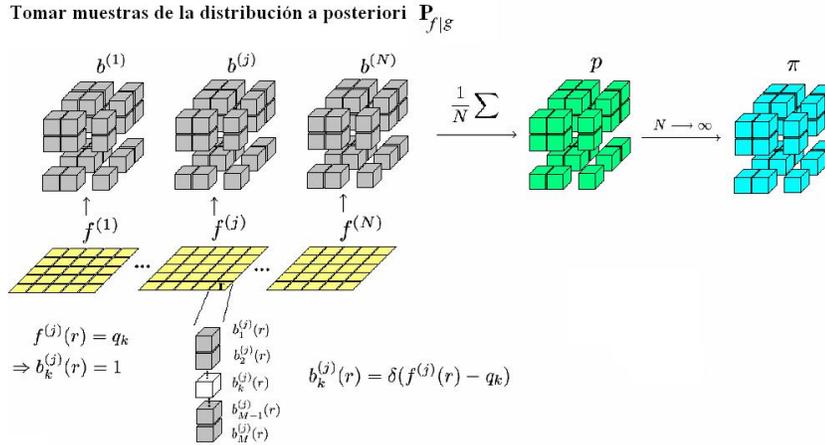


Figura 4.1: Método basado en Cadenas de Markov - Montecarlo(MCMC).

Un detalle muy importante es que aunque en teoría,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n f(X_t) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=T_0}^{T_0+n-1} f(X_t)$$

en la práctica (tenemos un número finito de muestras) utilizamos la segunda expresión. Las primeras muestras de la cadena $\{X_1, X_2, \dots, X_{T_0-1}\}$ se desechan porque no podemos asegurar que su distribución corresponde a la distribución a posteriori, así que, si n no es lo suficientemente grande, las marginales empíricas p no se aproximarán a las marginales verdaderas π .

El procedimiento anterior está fuertemente basado en la suposición de que contamos con una cadena de Markov con la distribución límite adecuada. El algoritmo Gibbs-Sampler (algoritmo 2), para construir una cadena de Markov adecuada, fue publicado por primera vez en 1984 por Stuart Geman y Donald Geman. Es fácil ver a partir del algoritmo que la cadena formada es regular y no periódica. Lo que no es tan fácil de ver es que la distribución asintótica es la distribución a posteriori. Una demostración de este teorema se encuentra en el trabajo original de Geman y Geman [16], se recomienda al lector interesado en este tema revisar este magnífico artículo.

Algorithm 2 Gibbs-Sampler

- 1: Elegir una configuración inicial b ;
 - 2: **for** $r \in \mathcal{S}$ **do**
 - 3: calcular $p_k(r) = P(b(r) = e_k, \bar{\theta} \mid g, b(s) \ s \in \mathcal{N}_r), k = 1, 2, \dots, M$;
 - 4: fijar $b(r) = e_k$ con probabilidad $p_k(r)$;
 - 5: **end for**
-

4.2. Campos de medida de Gauss-Markov

Los métodos basados en MCMC, descritos en la sección anterior, solo toman en cuenta la propiedad $E[p] = \pi$. Usando el teorema central del límite, podemos aprovechar una propiedad más de las marginales empíricas (figura 4.2)

$$\sqrt{n}(\hat{p} - \pi) \xrightarrow{n \rightarrow \infty} N(0, \Sigma)$$

donde

$$\Sigma = Cov(b) = E[b_i b_i^T] - E[b_i]E[b_i^T] = Diag(\pi) - \pi\pi^T$$

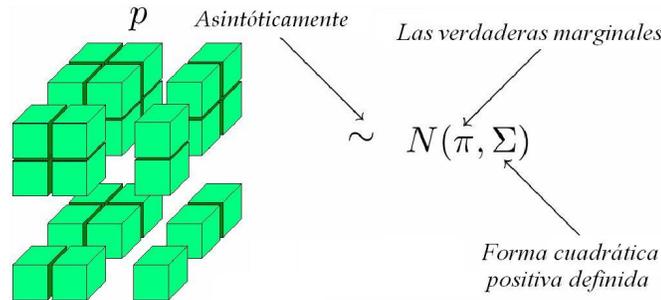


Figura 4.2: Modelo Campos de Medida de Gauss-Markov (GMMF) .

el problema, claro está, es que no conocemos π y por tanto no conocemos Σ . De cualquier forma, sabemos que la función de distribución asintótica de p está dada por

$$P(p) = \frac{1}{Z} \exp(-Q_{\Sigma}(p))$$

donde Z es una constante de normalización y Q_{Σ} es alguna función cuadrática positiva definida tal que $\nabla Q_{\Sigma}(\pi) = 0$. La idea en la que se basa el método GMMF es en modelar la función cuadrática Q_{Σ} . Para esto se introduce el concepto de “condición de consistencia” que tiene que ver con el hecho de que si no contamos con información a-priori, las muestras b fueron tomadas de la distribución dada por la verosimilitud, por lo que el valor esperado de p es la verosimilitud (normalizada) (la prueba de esto va mas allá del alcance de esta tesis). Una familia de funciones cuadráticas que cumplen con esta restricción y que fue propuesta como el modelo general GMMF, está dada por

$$Q(p) = \sum_{k=1}^M \sum_{r \in \mathcal{S}} (p_k(r) - v_k(s))^2 a_r + \sum_{\langle r, s \rangle \in \mathcal{C}_2} \sum_{k'=1}^M (p_k(r) - p_{k'}(s))^2 b(r, s, |k - k'|)$$

Un problema con este modelo es que las distribuciones estimadas $p(r)$ tienden a ser de alta entropía. En la figura 4.3 se muestra una comparación cualitativa del tipo de marginales obtenidas métodos

MCMC (Gibbs-Sampler), GMMF y EC-GMMF(este último método se describe más adelante). Lo que se grafica es el valor máximo de $p(r)$ en cada sitio r , los sitios claros se caracterizan por tener baja entropía y los sitios oscuros, alta.



Figura 4.3: Indicadores de entropía para diferentes marginales estimadas.

4.3. Campos Ocultos Markovianos de Medida (HMMF)

En las secciones anteriores se estudió un modelo de generación - observación a partir del cual formulamos el problema de segmentación de imágenes y concluimos con una solución de dicho problema que consiste en encontrar un campo de medida cuyos elementos denominamos *probabilidades marginales a posteriori*. En esta sección estudiaremos otro modelo de campo oculto que de igual forma nos sirve para formular el problema de segmentación, me referiré a este modelo como HMMF por *Hidden Markov Measure Field*. Usaremos un modelo de campo markoviano, por lo que la estructura del modelo HMMF se compone de un conjunto \mathcal{S} de sitios (píxeles de la imagen) y un sistema de vecindades \mathcal{N} el cual define un conjunto \mathcal{C} de *cliques*.

Sea N el numero de sitios en \mathcal{S} . Sea $\mathcal{S}_M \subset \mathbb{R}^M$ tal que $\forall x \in \mathcal{S}_M$.

$$\begin{cases} x_i \geq 0 & i = 1, 2, \dots, M \\ \sum_{i=0}^M x_i = 1 \end{cases}$$

La dinámica se compone de dos pasos, el modelo de generación y el modelo de observación. En el modelo de generación suponemos que existe una segmentación *verdadera* que fue generada mediante el modelo probabilístico ilustrado en la figura 4.4[1].

En el primer paso, un campo de probabilidades $p \in S_M^N$ (el cual modelaremos como un campo aleatorio Markoviano) es generado con probabilidad dada por la distribución de Gibbs

$$P(p) = \frac{1}{Z} \exp \left[- \sum_{C \in \mathcal{C}} W_C(p) \right]$$

donde W_C son funciones de potencial locales para los cliques $C \in \mathcal{C}$ y Z es una constante de normalización. Para cada sitio $r \in \mathcal{S}$ el vector $p(r)$ es una función de densidad discreta donde $p_k(r)$ es la probabilidad de que la etiqueta $f_r \in \mathcal{L} = \{q_1, q_2, \dots, q_M\}$ del sitio r sea q_k .

En un segundo paso, el campo de etiquetas f es generado de tal forma que cada etiqueta f_r es una muestra independiente con distribución $p(r)$.

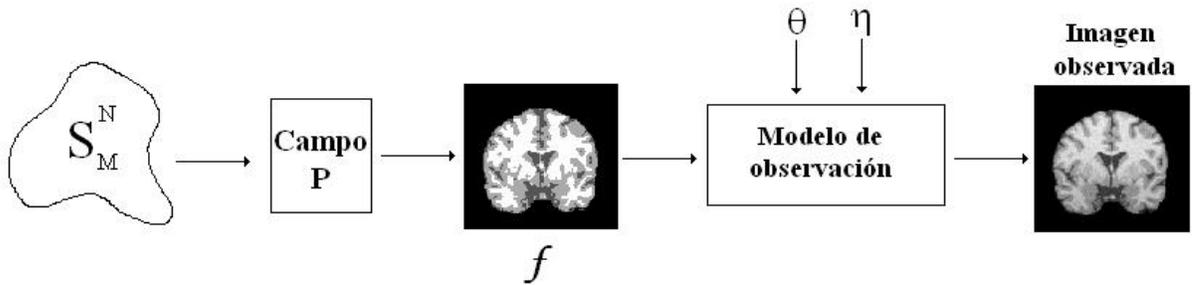


Figura 4.4: Modelo de generación - Observación.

Cada una de las M regiones (no necesariamente conexas) está caracterizada por algún rasgo (por ejemplo, el nivel de gris) que modelamos mediante una función paramétrica $\phi_i : \mathcal{S} \rightarrow \mathbb{R}^m$ con parámetros θ_i (una para cada región). Ejemplos de rasgos no escalares son el color y movimiento. Luego asumimos que la imagen verdadera fue generada simplemente evaluando para cada sitio, la función paramétrica correspondiente en el sitio dado.

La información con la que contamos para estimar los parámetros $\{\theta_i\}$ y el campo de probabilidades p es un conjunto de observaciones dado por una imagen $g : \mathcal{S} \rightarrow \mathbb{R}^d$ (nótese que g puede ser una imagen multibanda).

Denotemos por $v_k(r)$ para cada $r \in \mathcal{S}$ y cada $k = 1, 2, \dots, M$, la probabilidad $P(g(r) \mid f_r = q_k)$. Notemos que de acuerdo al modelo de generación-observación, el valor de $g(r)$ es independiente de p dado f_r , por lo que la probabilidad condicional conjunta

$$\begin{aligned}
P(g | p) &= \prod_{r \in \mathcal{S}} P(g(r) | p) = \prod_{r \in \mathcal{S}} \sum_{k=1}^K P(g(r) | f(r) = k) P(f(r) = k) = \\
&= \prod_{r \in \mathcal{S}} \sum_{k=1}^K v_k(r) p_k(r) = \prod_{r \in \mathcal{S}} v(r) \cdot p(r)
\end{aligned}$$

Estamos interesados en encontrar una expresión de la distribución *a-posteriori*

$$P(p | g) = \frac{P(g | p)P(p)}{P(g)} = \frac{1}{ZP(g)} \left[\prod_{r \in \mathcal{S}} v(r) \cdot p(r) \right] \exp(-V(p))$$

donde $V(p)$ es una función de potencial que define la distribución de Gibbs (*a-priori*) para p y depende de los cliques asociados al sistema de vecindades

$$V(p) = \sum_{C \in \mathcal{C}} W_C(p)$$

Para maximizar esta expresión respecto a p , tomamos el logaritmo natural

$$\log P(p | g) = c + \sum_{r \in \mathcal{S}} \log(v(r) \cdot p(r)) - \sum_{C \in \mathcal{C}} W_C(p)$$

lo cual es equivalente a minimizar

$$\sum_{r \in \mathcal{S}} -\log(v(r) \cdot p(r)) + \sum_{C \in \mathcal{C}} W_C(p)$$

la minimización del potencial está sujeta a las restricciones lineales

$$\begin{aligned}
\sum_{k=1}^M p_k(r) &= 1 \forall r \in \mathcal{S} \\
0 &\leq p_k(r) \leq 1
\end{aligned}$$

los autores de este método proponen minimizar U utilizando descenso newtoniano alternadamente sobre p y θ y para manejar las restricciones se propone proyectar los vectores $p(r)$ sobre el simplejo S_M [1]. Estos pasos de minimización -proyección tienen un costo computacional muy elevado además de que se requiere ajustar un conjunto de parámetros para la optimización respecto a p y otro conjunto para la optimización respecto a θ . Para acelerar la convergencia, se propone usar enfoque multiescala (*multi-scale approach*). El método comienza a ser cada vez más complicado debido al término “ $\log(p(r) \cdot v(r))$ ”.

Capítulo 5

Campos de Medida Markovianos cuadráticos con control de entropía

Science is wonderfully equipped to answer the question “How?” but it gets terribly confused when you ask the question “Why?”

Erwin Chargaff.

En esta sección se describe el modelo EC-QMMF (Entropy-Controlled Quadratic Markov Measure Fields), en la primera sección mostramos la derivación teórica, la cual termina con el planteamiento de un problema de optimización con restricciones. En la segunda sección se describe el método propuesto para resolver dicho problema de optimización. Finalmente en la tercera sección, mostramos una relación entre el campo de medida obtenido con EC-QMMF y las probabilidades marginales a posteriori que estudiamos en el capítulo anterior.

5.1. Derivación teórica

En la sección 3.4 planteamos el problema de segmentación como un problema de optimización combinatoria, en el que necesitamos minimizar cierta función de energía $U(d, \theta)$ respecto a las variables binarias b , las cuales modelamos como un campo aleatorio Markoviano. Precisamente el hecho de que b toma valores en un conjunto discreto es lo que dificulta esta tarea. Nos gustaría poder derivar U respecto a b para utilizar métodos de optimización más eficientes. Para hacer esto, permitiremos que b tome

valores en un continuo, reemplazaremos las condiciones que impusimos al principio por las condiciones relajadas

$$b_k(r) \geq 0 \forall 1 \leq k \leq M, r \in \mathcal{S} \quad (5.1)$$

$$\sum_{k=1}^M b_k(r) = 1 \forall r \in \mathcal{S} \quad (5.2)$$

$$b_k(r) \approx \delta(k - c_r) \quad (5.3)$$

donde c_r es la clase a la que pertenece r .

Como se explicó en la sección 3.4, deseamos que b sea suave, pero esta vez, al tomar valores en un continuo podemos usar potenciales más generales, diferenciables, en lugar del potencial de Ising. Una elección sencilla es el potencial cuadrático visto en la sección 3.1. La siguiente función de energía expresa estas propiedades

$$U_b(b) = \lambda \sum_{\langle r,s \rangle \in \mathcal{C}_2} \|b(r) - b(s)\|^2 - \mu \sum_{r \in \mathcal{S}} \sum_{k=1}^M b_k(r)^2 \quad (5.4)$$

El primer término corresponde a la condición de suavidad. El segundo término (un factor del índice de Gini menos una constante) penaliza la entropía de los vectores $b(r)$ (debido a las restricciones (5.1) y (5.2)). Por lo anterior, b es un campo aleatorio de medida (por las condiciones (5.1) y (5.2)) Markoviano con control de entropía. Este control lo obtenemos al variar el parámetro no negativo μ .

Si suponemos que las variables aleatorias η_r (ver sección 3.4) son independientes e idénticamente distribuidas con distribución normal con media 0 y varianza σ^2 , entonces la verosimilitud de la imagen observada (usando el mismo modelo de la sección 3.4) es

$$P(g | b, \theta) = \frac{1}{Z_l} \exp \left[-\frac{1}{2\sigma^2} \sum_{r \in \mathcal{S}} \left(g(r) - \sum_{i=1}^M b_i(r) \phi_{\theta_i} \right)^2 \right]$$

usando la regla de Bayes, la distribución *a posteriori* es

$$P(b | g, \theta) = \frac{1}{Z} \exp[-U(b, \theta)]$$

donde Z es una constante de normalización y

$$U(b, \theta) = \sum_{r \in \mathcal{S}} \left[\frac{1}{2\sigma^2} \left(g(r) - \sum_{i=1}^M b_i(r) \phi_{\theta_i}(r) \right)^2 - \mu \sum_{k=1}^M b_k(r)^2 \right] + \lambda \sum_{\langle r,s \rangle \in \mathcal{C}_2} \|b(r) - b(s)\|^2$$

Notemos que por (5.2)

$$g(r) - \sum_{k=1}^M b_k(r) \phi_{\theta_k}(r) = \sum_{k=1}^M b_k(r) (g(r) - \phi_{\theta_k}(r))$$

y en el límite de baja entropía se cumple la condición (5.3), así que

$$\left[\sum_{k=1}^M b_k(r) (g(r) - \phi_{\theta_k}(r)) \right]^2 \approx \sum_{k=1}^M b_k(r)^2 (g(r) - \phi_{\theta_k}(r))^2$$

usando esta aproximación podemos escribir la función de energía como

$$U(b, \theta) = \frac{1}{2\sigma^2} \sum_{r \in \mathcal{S}} \sum_{i=1}^M b_i(r)^2 [(g(r) - \phi_{\theta_i}(r))^2 - \mu] + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} \|b(r) - b(s)\|^2 \quad (5.5)$$

El estimador MAP para b se encuentra minimizando (5.5) respecto a b , sujeta a las restricciones (5.1) y (5.2).

Para obtener una expresión más general, sin asumir que las variables η_r (el ruido) tienen distribución Gaussiana, consideremos como en el modelo de campos ocultos (HMMF) visto al final del capítulo anterior, el siguiente proceso de dos pasos: en el primer paso un campo aleatorio Markoviano de medida b se genera con distribución de Gibbs con función de energía (5.4), en el segundo paso la etiqueta f_r de cada sitio r es elegida de manera independiente con distribución $b(r)$, de tal forma que $P(f_r = q_k) = b_k(r)$. Definamos, como antes, $v_k(r) = P(g(r) \mid f_r = q_k)$. Como b y g son independientes dado f , tenemos que $v_k(r) = P(g(r) \mid f_r = q_k, b)$, entonces

$$P(g(r) \mid b, \theta) = \sum_{k=1}^M P(g(r) \mid f_r = q_k, b, \theta) P(f_r = q_k) = \sum_{k=1}^M v_k(r) b_k(r)$$

usando, como antes, la regla de Bayes obtenemos la función de energía

$$U(b, \theta) = \sum_{r \in \mathcal{S}} \left[-\log \left(\sum_{k=1}^M v_k(r) b_k(r) \right) - \mu \sum_{k=1}^M b_k(r)^2 \right] + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} \|b(r) - b(s)\|^2 \quad (5.6)$$

pero en el límite de baja entropía, podemos usar la siguiente aproximación

$$\log \left(\sum_{k=1}^M v_k(r) b_k(r) \right) \approx \sum_{k=1}^M b_k(r)^2 \log(v_k(r))$$

La función de energía general es entonces

$$U(b, \theta) = \sum_{r \in \mathcal{S}} \sum_{k=1}^M b_k(r)^2 (-\log(v_k(r)) - \mu) + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} \sum_{k=1}^M (b_k(r) - b_k(s))^2$$

la cual, en el caso particular de ruido Gaussiano, es (5.5) si consideramos que la constante $\frac{1}{2\sigma^2}$ puede incorporarse en los parámetros μ, λ (basta multiplicar toda la expresión por $2\sigma^2$). Notemos que en el caso general, la energía sigue siendo cuadrática respecto a b , es por esto que llamamos a este modelo campo de medida markoviano cuadrático con control de entropía (EC-QMMF por *Entropy Controlled Quadratic Markov Measure Field*).

5.2. Método propuesto de optimización con restricciones

El método de los multiplicadores de Lagrange [30] para optimización con restricciones consiste en encontrar un punto crítico de la *función lagrangiana*. Consideremos el problema de optimización con restricciones definido por

$$\min_{x \in \mathbb{R}^n} f(x)$$

sujeto a

$$c_i(x) \geq 0, i \in \mathcal{I}$$

$$c_i(x) = 0, i \in \mathcal{E}$$

donde \mathcal{I} es el conjunto de restricciones de desigualdad y \mathcal{E} es el conjunto de restricciones de igualdad. La función lagrangiana asociada a este problema es

$$\mathcal{L}[f, c, \lambda](x) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$$

Las condiciones de Karush-Kuhn-Tucker, establecen que si x es un mínimo local de f en la región factible definida por las restricciones y los gradientes $\{\nabla c_i(x)\}$ son linealmente independientes, entonces existe un conjunto de multiplicadores de Lagrange $\{\lambda_i\}$ tales que

$$\nabla_x \mathcal{L}[f, c, \lambda](x) = 0$$

$$c_i(x) = 0 \quad \forall i \in \mathcal{E}$$

$$c_i(x) \geq 0 \quad \forall i \in \mathcal{I}$$

$$\lambda_i \geq 0 \quad \forall i \in \mathcal{I}$$

$$\lambda_i c_i(x) = 0 \quad \forall i \in \mathcal{I}$$

Muchos métodos han sido desarrollados en base a estas condiciones, pero la idea básica detrás de ellos es encontrar de manera eficiente un punto crítico de \mathcal{L} que las satisfaga. A continuación describimos el método utilizado en este trabajo.

La función de energía en la que estamos interesados es $U(p, \theta)$, donde p es un campo de probabilidades y θ es el conjunto de parámetros de los modelos. Las restricciones son naturalmente, las que definen a p como un campo de medida. Por ahora consideremos únicamente las restricciones de igualdad

$$\sum_i p_i(r) = 1 \quad \forall r \in \mathcal{S} \quad (5.7)$$

para hacer las cuentas un poco mas limpias usaremos las restricciones de igualdad equivalentes $2c_i(x) = 0$, es decir $2 \sum_i p_i(r) = 2$, luego la función Lagrangiana es

$$\mathcal{L}[U, \pi](p, \theta) = U(p, \theta) - \sum_{r \in \mathcal{S}} 2\pi_r \left[\sum_{k=1}^M p_k(r) - 1 \right]$$

donde π_r es el multiplicador de Lagrange asociado a la restricción de igualdad para el sitio r . Para encontrar un punto crítico de \mathcal{L} necesitamos resolver el sistema de ecuaciones dado por

$$\frac{\partial \mathcal{L}}{\partial p_k(r)} = \frac{\partial U}{\partial p_k(r)} - 2\pi_r = 0 \quad (5.8)$$

por otro lado

$$\frac{\partial U}{\partial p_k(r)}(p, \theta) = -2p_k(r)(\log(v_k(r)) + \mu) + 2\lambda \sum_{s \in \mathcal{N}_r} (p_k(r) - p_k(s))$$

sustituyendo en (5.8) y resolviendo para $p_k(r)$, tenemos

$$p_k(r) = \frac{\pi_r + \lambda \sum_{s \in \mathcal{N}_r} p_k(s)}{-\log v_k(r) - \mu + \lambda \#\mathcal{N}_r} \quad (5.9)$$

donde \mathcal{N}_r es el conjunto de sitios $s \in \mathcal{S}$ que son vecinos de r .

Sean

$$\begin{aligned} n_k(r) &:= \lambda \sum_{s \in \mathcal{N}_r} p_k(s) \\ m_k(r) &:= -\log v_k(r) - \mu + \lambda \#\mathcal{N}_r \end{aligned}$$

donde $\#\mathcal{N}_r$ denota la cardinalidad de \mathcal{N}_r . Sustituyendo (5.9) en (5.7) y usando esta notación obtenemos

$$\sum_k \frac{\pi_r + n_k(r)}{m_k(r)} = 1 \Rightarrow \pi_r = \frac{1 - \sum_i \frac{n_i(r)}{m_i(r)}}{\sum_i \frac{1}{m_i(r)}}$$

finalmente, sustituyendo en (5.9) obtenemos

$$p_k(r) = \frac{n_{k,r}}{m_{k,r}} + \frac{1 - \sum_{i=1}^M \frac{n_{i,r}}{m_{i,r}}}{\sum_{i=1}^M \frac{1}{m_{i,r}}} \quad (5.10)$$

Hay que recordar que esta expresión es la solución del problema de optimización con restricciones de igualdad (ignoramos las restricciones de desigualdad $\{0 \leq p_k(r) \leq 1\}$). Lo cual significa que esta solución solamente asegura que $\sum_i p_i(r) = 1 \forall r \in \mathcal{S}$. Experimentalmente un procedimiento que funciona adecuadamente es *proyectar* cada variable $p_k(r)$ haciendo

$$p_k(r) = \max\{p_k(r), 0\}$$

Después de esta asignación ya no podemos asegurar que se cumple la restricción $\sum_i p_i(r) = 1$, así que es necesario un segundo paso que consiste en una *normalización* haciendo

$$p_k(r) = \frac{p_k(r)}{\sum_i p_i(r)}$$

Notemos que la expresión (5.10) está en términos de $\log v_k(r)$, lo cual sugiere que lo anterior puede ser utilizado en cualquier contexto en el que contemos con dos componentes básicos: un modelo para la imagen *verdadera* (f) y un modelo probabilístico de la imagen observada (g).

Recordemos que además del campo de probabilidades p estamos interesados en calcular los parámetros θ_i de ϕ_i , por lo que es deseable que $\log v_k(r)$ sea fácil de optimizar respecto a θ , es por esto que la elección

$$v_k(r) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(g(r) - f(r))^2}{2\sigma^2}\right)$$

es muy conveniente, ya que

$$\log v_k(r) = c - \frac{1}{2\sigma^2}(g(r) - f(r))^2$$

Algunos ejemplos de modelos que podrían ser útiles en diferentes problemas de visión computacional de bajo nivel son

- $\phi_i(r) = \mu_i \in \mathbb{R} \forall r \in \mathcal{S}$, i.e., modelos constantes, útiles en situaciones en las que las imágenes son relativamente sencillas, puede existir alto nivel de ruido pero no son significativas las variaciones espaciales en los modelos debidas a artefactos de iluminación o contraste. Ejemplos de rasgos que se representan con estos modelos (escalares) son el nivel de gris y la disparidad estereoscópica.
- $\phi_i(r) = \mu_i \in \mathbb{R}^2 \forall r \in \mathcal{S}$, i.e. modelos constantes bidimensionales que pueden representar desplazamientos en secuencias de imágenes. Esta clase de modelos pueden ser utilizados en segmentación de movimiento o cálculo de flujo óptico en secuencias de imágenes en las que los desplazamientos de interés son relativamente simples tales como traslaciones.

- $\phi_i(r) = \sum_j \theta_j^{(i)} N_j^{(i)}(r)$, donde $\{N_j\}$ es un conjunto de “funciones base”, como splines de producto tensorial. Esta clase de modelos permiten variaciones espaciales suaves.
- $\phi_i(r) = f_i : \mathbb{R} \rightarrow \mathbb{R}$, donde $\{f_i\}$ es un conjunto de funciones que podrían ser por ejemplo modelos de series de tiempo. Son útiles en situaciones en las que la observación $g(r)$ es de hecho un conjunto de observaciones (por ejemplo una sucesión de valores tomados en un mismo sitio a lo largo de un periodo de tiempo)

Mas adelante mostramos aplicaciones de los modelos anteriores desarrollando para cada uno, el método concreto de estimación de parámetros (notemos que esto es lo único que varía ya que el cálculo del campo p es idéntico una vez calculada $\log v_k(r)$ para cada sitio $r \in \mathcal{S}$).

En general el paso de estimación de parámetros consiste en resolver para θ el problema de optimización dado por

$$\theta^* = \operatorname{argmin}_{\theta} U(p, \theta) \quad (5.11)$$

El algoritmo completo es el siguiente

Algorithm 3 Método de Gauss-Seidel (GS) para el cálculo del campo de medida EC-QMMF

- 1: Elegir el número M de modelos
 - 2: Fijar valores iniciales para los parámetros θ_0 e inicializar $p_k^{(0)}(r) = \frac{1}{M} \forall r \in \mathcal{S}, 1 \leq k \leq M$
 - 3: Fijar la tolerancia $\epsilon > 0$, y un número máximo de iteraciones T .
 - 4: **for** $t = 1, 2, \dots, T$ **do**
 - 5: **for all** $r \in \mathcal{S}$ **do**
 - 6: **for all** $1 \leq k \leq M$ **do**
 - 7: Calcular $p_k^{(t)}(r)$ con (5.10)
 - 8: Proyectar $p_k^{(t)}(r) \leftarrow \max\{0, p_k^{(t)}(r)\}$
 - 9: **end for** $1 \leq k \leq M$
 - 10: Renormalizar cada vector $p(r), r \in \mathcal{S}$
 - 11: **end for** $r \in \mathcal{S}$
 - 12: Actualizar los modelos θ_t resolviendo (5.11)
 - 13: **if** $\|p^{(t)} - p^{(t-1)}\| < \epsilon$ **then**
 - 14: PARAR con solución $p_k^*(r) = p_k^{(t)}(r)$ y $\theta^* = \theta_t$
 - 15: **end if**
 - 16: **end for**
-

Para evitar el problema de escala debido al tamaño de la imagen de entrada y el número de modelos,

utilizamos la norma

$$\|p\| = \sqrt{\frac{1}{z} \sum_{r \in \mathcal{S}} \frac{1}{M} \sum_{k=1}^M p_k(r)^2}$$

donde z es el número píxeles de la imagen y M es el número de modelos.

5.3. Campos de Medida de Gauss-Markov con entropía controlada (EC-GMMF)

En la publicación original de GMMF, se probó que si el campo de etiquetas f es un campo Markoviano respecto a un sistema de vecindades \mathcal{N} , entonces el campo de *probabilidades marginales empíricas* p es también un campo aleatorio Markoviano respecto al mismo sistema de vecindades [28]. Al final las *verdaderas* marginales fueron aproximadas tomando el valor esperado de p , el cual, al ser p de distribución normal, *debe* ser el punto crítico de una forma cuadrática, positiva definida, de p . Para restringir el espacio de las posibles formas cuadráticas a aquellas que potencialmente corresponden a la que nos interesa, se estudió el caso en que no se cuenta con información a priori de f . En este caso podemos suponer el sistema de vecindades degenerado (la 0-vecindad) para f , con lo cual

$$P(f | g) = \frac{P(g | f)P(f)}{P(g)} \propto P(g | f)$$

por lo tanto

$$\pi_k(r) = P(f_r = q_k | g_r) = \frac{P(g_r | f_r = q_k)}{\sum_{i=1}^M P(g_r | f_r = q_i)} = \frac{v_k(r)}{\sum_{i=1}^M v_i(r)}$$

la verosimilitud normalizada. A partir de esta restricción, denominada *condición de consistencia* se propuso una *forma general* del potencial Q_Σ .

Para explorar las propiedades de otras formas cuadráticas como modelos de Q_Σ es necesario *relajar* esta condición de consistencia. Una idea razonable es buscar que el estimador óptimo p^* cumpla

$$\operatorname{argmax}_k p_k^*(r) = \operatorname{argmax}_k v_k(r) \quad \forall r \in \mathcal{S}$$

es decir, el estimador MPM coincide con el estimador de máxima verosimilitud [33]. Además, para resolver el problema de alta entropía que presentan las marginales estimadas mediante GMMF, sería deseable un parámetro que mantenga dicha entropía controlada.

Una familia de funciones que cumplen con esta propiedad está dada por la aproximación cuadrática del potencial que surge del modelo de campos ocultos, descrito en la sección anterior

$$U(p) = \sum_{r \in \mathcal{S}} \sum_{k=1}^M -p_k(r)^2 \log(v_k(r)) + \lambda \sum_{\langle r,s \rangle \in \mathcal{C}_2} \sum_{k=1}^M (p_k(r) - p_k(s))^2 - \mu \sum_{r \in \mathcal{S}} \sum_{k=1}^M p_k(r)^2$$

el último término es el que controla la entropía de las distribuciones $p(r)$, se trata de una penalización del coeficiente de Gini

$$G(p(r)) = 1 - \sum_{k=1}^M p_k(r)^2$$

Lo anterior muestra una relación interesante entre el modelo de campos ocultos HMMF y el de Gauss-Markov GMMF que bien valdría la pena estudiar con mayor profundidad.

Capítulo 6

Resultados experimentales

No amount of experimentation can ever prove me right; a single experiment can prove me wrong.

Albert Einstein.

En adelante asumiremos que la distribución de η_r es normal.

Para mostrar los resultados experimentales comenzaremos con el modelo de ϕ_i más sencillo, el de modelos constantes $\phi_i(r) = \theta_i \forall r \in \mathcal{S}$. De este modo, la función de energía es

$$U(p, \theta) = \sum_{r \in \mathcal{S}} \sum_i \left(\frac{(\theta_i - g_r)^2}{2\sigma^2} - \mu - \log(\sqrt{2\pi}\sigma) \right) p_i(r)^2 + \lambda \sum_i \sum_{\langle r, s \rangle \in \mathcal{S}} (p_i(r) - p_i(s))^2$$

El valor de σ puede ser manejado de dos maneras distintas, una es simplemente multiplicar la expresión anterior por $2\sigma^2$ para obtener

$$U(p, \theta) = \sum_{r \in \mathcal{S}} \sum_i ((\theta_i - g_r)^2 - 2\sigma^2(\mu + \log(\sqrt{2\pi}\sigma))) p_i(r)^2 + 2\sigma^2 \lambda \sum_i \sum_{\langle r, s \rangle \in \mathcal{S}} (p_i(r) - p_i(s))^2$$

y dado que λ es un parámetro del método podemos ignorar el factor σ y suponer que el parámetro a ajustar es simplemente $\hat{\lambda} = 2\sigma^2\lambda$, de la misma manera podemos incorporar el término $\log(\sqrt{2\pi}\sigma)$ en el parámetro $\hat{\mu} = 2\sigma^2(\mu + \log(\sqrt{2\pi}\sigma))$ lo cual nos lleva a un método que requiere del ajuste de dos parámetros, $\hat{\lambda}$ y $\hat{\mu}$, que en general serán diferentes para diferentes niveles de ruido. La función de energía final es (6.1).

$$U(p, \theta) = \sum_{r \in \mathcal{S}} \sum_i ((\theta_i - g(r))^2 - \hat{\mu}) p_i(r)^2 + \hat{\lambda} \sum_i \sum_{\langle r, s \rangle \in \mathcal{S}} (p_i(r) - p_i(s))^2 \quad (6.1)$$

Otra manera de proceder es fijar $\lambda = c$, alguna constante determinada a partir de un conjunto de imágenes de prueba (elegir c como el valor con el que se obtienen los mejores resultados) y estimar σ a partir de los datos. Esto puede realizarse de diferentes maneras, en este trabajo estudiamos en particular las siguientes opciones:

- Estimación de σ por ventanas: consiste en estimar la varianza local a partir de la imagen original fijando para cada sitio $r \in \mathcal{S}$ una ventana $W_r \subset \mathcal{S}$, cada ventana define una estimación local de la varianza

$$\sigma_r^2 = \frac{1}{\|W_r\|} \sum_{s \in W_r} (g(s) - m_r)^2$$

donde

$$m_r = \frac{1}{\|W_r\|} \sum_{s \in W_r} g(s)$$

Finalmente dado el conjunto de estimadores locales $\{\sigma_r\}$, obtenemos un histograma y elegimos como estimador de la varianza la moda de dicho histograma. Este procedimiento resulta ser muy robusto ya que las variaciones en los estimadores σ_r debidas por ejemplo a los bordes de las imágenes no tienen un efecto importante sobre el estimador elegido (la moda).

- Estimación de σ a partir del modelo original: consiste en considerar σ como un parámetro más que debe ser estimado minimizando la energía. En este caso debemos considerar todos los términos de $\log v_k(r)$:

$$\frac{\partial U}{\partial \sigma}(p, \theta) = \sum_r \sum_k \left(\frac{(\theta_k - g(r))^2}{\sigma^3} - \frac{1}{\sigma} \right) p_k(r)^2$$

multiplicando por σ^3 tenemos

$$\frac{\partial U}{\partial \sigma}(p, \theta) = 0 \Rightarrow \sigma^2 = \frac{\sum_r \sum_k p_k(r)^2 (\theta_k - g(r))^2}{\sum_r \sum_k p_k(r)^2} \quad (6.2)$$

Análogamente para obtener el estimador óptimo para θ_k

$$\frac{\partial U}{\partial \theta_k} = \sum_{r \in \mathcal{S}} \frac{(\theta_k - g(r))}{\sigma^2} p_k(r)^2 = 0 \Rightarrow \theta_k = \frac{\sum_{r \in \mathcal{S}} g(r) p_k(r)^2}{\sum_{r \in \mathcal{S}} p_k(r)^2} \quad (6.3)$$

Ambos procedimientos presentan resultados similares. Sin embargo, es importante que se utilice un σ común para todos los modelos, es decir, los resultados experimentales son bastante malos si definimos

la verosimilitud usando σ_k para el modelo k y actualizamos cada una por separado.

A continuación mostramos los resultados obtenidos en la segmentación de imágenes sintéticas. Se comparó el método propuesto (EC-QMMF) contra el modelo original de campos ocultos (HMMF) y el estimador MAP para el modelo mostrado en la sección 3.4. Como mostramos en dicha sección, la dificultad de este modelo es que la función de energía resultante de este modelo debe minimizarse respecto a un conjunto de variables discretas. Entre los métodos más efectivos para atacar este problema de optimización combinatoria destacan los algoritmos basados en cortes de grafos (GC por *Graph-Cuts*). En la siguiente sección se describen estos métodos, sin embargo se recomienda revisar la literatura para un estudio más detallado [5][6][23], los resultados que mostramos mas adelante se obtuvieron siguiendo las indicaciones de estos artículos.

El estudio comparativo se divide en dos etapas. En la primera etapa (exploración) se eligen parámetros adecuados para los diferentes métodos, en la siguiente etapa (validación) se utilizan los parámetros elegidos para comparar el desempeño de los tres métodos al usarlo para segmentar imágenes con diferentes niveles de ruido.

6.1. Graph-cuts

Estamos interesados en minimizar funciones de energía de la forma

$$E(f) = \sum_{p \in \mathcal{S}} D_p(f_p) + \sum_{\langle p, q \rangle \in \mathcal{C}_2} V_{\langle p, q \rangle}(f_p, f_q) \quad (6.4)$$

donde f es un campo de etiquetas definido sobre el conjunto de sitios \mathcal{S} con valores en $\mathcal{L} = \{q_1, q_2, \dots, q_M\}$, D_p son funciones no negativas arbitrarias que corresponden al *término de datos* y $V_{\langle p, q \rangle}$ son funciones no negativas que modelan la información *a priori*, típicamente corresponden al *término de regularización* para obtener campos de etiquetas *suaves a pedazos*¹. Greig, Porteous, y Seheult [17] mostraron un método basado en cortes de grafos para encontrar el mínimo global de la función de energía para el caso de dos modelos (segmentación binaria). En 2001, Yuri Boykov y colaboradores presentaron dos algoritmos de optimización muy eficientes también basados en cortes de grafos para el caso de más de

¹En la expresión 6.4 asumimos que $\langle p, q \rangle$ es un clique de orden 2 para hacer más sencilla la relación de estas funciones de energía con el problema de segmentación de imágenes, pero en realidad puede ser cualquier par de sitios y el tipo de funciones V puede ser aun mas general. Se invita al lector a revisar el artículo de Boykov [5].

dos modelos: expansión α (α expansion) e intercambio $\alpha - \beta$ ($\alpha - \beta$ swap). Aunque no pueden ser usados para minimizar una función de energía arbitraria ni aseguran la convergencia a un mínimo global, estos algoritmos encuentran eficientemente *buenos* mínimos locales para un conjunto muy general de funciones de energía[5].

Definición: dada una etiqueta $\alpha \in \mathcal{L}$, un movimiento de un etiquetado f a un nuevo etiquetado f' es una *expansión α* (α -Expansion), si para todo r se cumplen las siguientes condiciones

$$\begin{aligned} f_r = \alpha &\Rightarrow f'_r = \alpha \\ f'_r \neq \alpha &\Rightarrow f'_r = f_r \end{aligned}$$

Definición: dado un par de etiquetas $\alpha, \beta \in \mathcal{L}$, un movimiento de un etiquetado f a un nuevo etiquetado f' es un *intercambio $\alpha - \beta$* ($\alpha - \beta$ Swap), si para cualquier etiqueta $\gamma \neq \alpha, \beta$, se cumple que $f_r = f'_r$.

Algorithm 4 Expansión α

```

1: Elegir un campo de etiquetas inicial  $f$ ;
2: huboCambios:=true;
3: while huboCambios do
4:   huboCambios:=false;
5:   for  $\alpha \in \mathcal{L}$  do
6:     Encontrar  $\hat{f} = \operatorname{argmin} E(f')$  para todas las  $f'$  que puedan ser obtenidas mediante una expansión  $\alpha$  (ver texto) partiendo de  $f$ ;
7:     if  $E(\hat{f}) < E(f)$  then
8:        $f := \hat{f}$ ;
9:       huboCambios:=true;
10:    end if
11:  end for
12: end while

```

El algoritmo α -expansion (algoritmo 4) requiere que los potenciales $V_{\langle p,q \rangle}$ sean *normas* y el algoritmo $\alpha - \beta$ swap (algoritmo 5) solo requiere que sean *semi normas*. Una norma es una función que satisface las siguientes condiciones

1. $V(\alpha, \beta) = V(\beta, \alpha) \geq 0$
2. $V(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$
3. $V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \beta)$

Algorithm 5 Intercambio $\alpha - \beta$

```

1: Elegir un campo de etiquetas inicial  $f$ ;
2: huboCambios:=true;
3: while huboCambios do
4:   huboCambios:=false;
5:   for cada par  $\alpha, \beta \in \mathcal{L}$  do
6:     Encontrar  $\hat{f} = \operatorname{argmin} E(f')$  para todas las  $f'$  que puedan ser obtenidas mediante un intercambio  $\alpha - \beta$  (ver texto) partiendo de  $f$ ;
7:     if  $E(\hat{f}) < E(f)$  then
8:        $f := \hat{f}$ ;
9:       huboCambios:=true;
10:    end if
11:  end for
12: end while

```

para cualesquiera etiquetas $\alpha, \beta, \gamma \in \mathcal{L}$. Si V solamente satisface las primeras dos condiciones, decimos que es una seminorma.

El paso mas importante de ambos algoritmos es el de la linea 6. El mínimo global respecto a los movimientos permitidos (ya sea expansiones α o intercambios $\alpha - \beta$) se encuentra de manera exacta construyendo adecuadamente un grafo con dos terminales, de tal forma que el mínimo corte que separa las terminales define el etiquetado óptimo que se necesita en la linea 6 de ambos algoritmos (en [5] se describe la construcción del grafo adecuado para expansiones α y del grafo adecuado para intercambios $\alpha - \beta$). En 2004, Yuri Boykov y Vladimir Kolmogorov publicaron una implementación muy eficiente de máximo flujo/mínimo corte para grafos arbitrarios [7]. Esta implementación es la que utilizamos en esta comparación y se encuentra disponible en la página personal de Vladimir Kolmogorov ².

En [5] se contempla el uso de potenciales robustos para preservar los bordes de las imágenes como mencionamos en la sección 3.1, siempre y cuando el potencial utilizado sea una norma o seminorma. En dicho artículo, el conjunto de etiquetas \mathcal{L} es el conjunto de todas las intensidades o colores posibles para cada pixel, por lo que no es necesario un paso de estimación de parámetros como el que sugerimos en esta tesis, sin embargo para hacer una comparación justa de este método con HMMF y EC-QMMF, utilizaremos versiones de dos pasos de los algoritmos α -Expansion y $\alpha - \beta$ Swap al estilo del algoritmo SP (ver algoritmos 6 y 7). Utilizaremos modelos constantes $\mu = \{\mu_1, \mu_2, \dots, \mu_M\}$ donde M es el número de modelos utilizados.

²<http://www.adastral.ucl.ac.uk/vladkolm/software.html>

Algorithm 6 Algoritmo SP (Para encontrar el estimador MAP del problema de segmentación paramétrica usando α -Expansion)

```

1: Elegir los modelos iniciales  $\mu^{(0)}$  distribuyéndolos uniformemente en el rango dinámico de la imagen
   observada  $g$ ;
2: Inicializar el campo de etiquetas  $f_r = q_k$  tal que  $k = \operatorname{argmin}_k (|g_r - \mu_k^{(0)}|)$ ;
3: Fijar un número máximo de iteraciones  $T > 0$ ;
4:  $t:=1$ ;
5: huboCambios:=true;
6: while huboCambios y  $t \leq T$  do
7:   huboCambios:=false;
8:   Actualizar  $\mu_k^{(t)} = \frac{1}{N_k} \sum_{r:f_r=k} g_r$ , donde  $N_k$  es el número de pixeles asociados a la clase  $k$  en  $f^{(t)}$ ;
9:   for  $\alpha \in \mathcal{L}$  do
10:    Encontrar  $\hat{f} = \operatorname{argmin} E(f')$  para todas las  $f'$  que puedan ser obtenidas mediante una expansión
       $\alpha$  (ver texto) partiendo de  $f$ ;
11:    if  $E(\hat{f}) < E(f)$  then
12:       $f := \hat{f}$ ;
13:      huboCambios:=true;
14:    end if
15:  end for
16:   $t:=t+1$ ;
17: end while

```

Un punto muy importante es que en [5] se recomienda el uso del potencial de Hampel (ver sección 3.1.1 de esta tesis) en el término de datos, así que para las tres primeras funciones de energía utilizaremos este potencial en el término de datos. En el mismo artículo se mencionan como posibles potenciales de regularización el de Ising, la norma L_1 y el potencial de Hampel (ver secciones 3.4 y 3.1.1), los dos primeros son normas, por lo que usaremos el algoritmo α -Expansion y el tercero es una semi norma por lo que usaremos el algoritmo $\alpha - \beta$ Swap. En [34] se sugiere utilizar el potencial de Andrew dado por

$$\rho(x) = 1 - \exp(-kx^2)$$

tanto en el término de datos como en el término de regularización, como veremos en seguida se obtienen mejores resultados que al usar los potenciales sugeridos en [5]. El potencial de Andrew es una seminorma por lo que usaremos el algoritmo $\alpha - \beta$ Swap para minimizar la energía asociada a este potencial.

Algorithm 7 Algoritmo SP (Para encontrar el estimador MAP del problema de segmentación paramétrica usando $\alpha - \beta$ Swap)

```

1: Elegir los modelos iniciales  $\mu^{(0)}$  distribuyéndolos uniformemente en el rango dinámico de la imagen
   observada  $g$ ;
2: Inicializar el campo de etiquetas  $f_r = q_k$  tal que  $k = \operatorname{argmin}_k (|g_r - \mu_k^{(0)}|)$ ;
3:  $t:=1$ ;
4: huboCambios:=true;
5: while huboCambios y  $t \leq T$  do
6:   huboCambios:=false;
7:   Actualizar  $\mu_k^{(t)} = \frac{1}{N_k} \sum_{r: f_r=k} g_r$ , donde  $N_k$  es el número de pixeles asociados a la clase  $k$  en  $f^{(t)}$ ;
8:   for cada par  $\alpha, \beta \in \mathcal{L}$  do
9:     Encontrar  $\hat{f} = \operatorname{argmin} E(f')$  para todas las  $f'$  que puedan ser obtenidas mediante un intercambio
        $\alpha - \beta$  (ver texto) partiendo de  $f$ ;
10:    if  $E(\hat{f}) < E(f)$  then
11:       $f := \hat{f}$ ;
12:      huboCambios:=true;
13:    end if
14:  end for
15:   $t:=t+1$ ;
16: end while

```

6.2. Comparación

Exploraremos una región del espacio de parámetros de cada función de energía en la que “seguramente” se encuentra el conjunto óptimo para una imagen dada. La región de búsqueda se fija empíricamente observando el comportamiento de cada función de energía al variar cada parámetro. En el caso de los estimadores MAP, para todas las funciones de energía observamos que es posible encontrar buenas soluciones eligiendo cada parámetro en algún valor del intervalo $[0.1, 1]$. En el caso de HMMF la región de búsqueda es $(\lambda, \sigma) \in [0.5, 2] \times [0.5, 4]$.

La siguiente tabla muestra los potenciales utilizados en las cuatro funciones de energía para el estimador MAP, también muestra las regiones exploradas del espacio de parámetros y el número de pruebas hechas en dicha exploración.

Potenciales usados en el estudio comparativo con el estimador MAP					
Nombre	$D(x)$	$V(x, y)$	Parámetros	Región de búsqueda	Núm. pruebas
Hampel-Ising	$\min(x^2, k_1)$	$\lambda \cdot \delta(x - y)$	k_1, λ	$[0.1, 1] \times [0.1, 1]$	20×20
Hampel- L_1	$\min(x^2, k_1)$	$\lambda \cdot \min(\ x - y\ , k_2)$	k_1, λ, k_2	$[0.1, 1] \times [0.1, 1] \times [0.1, 1]$	$10 \times 10 \times 10$
Hampel-Hampel	$\min(x^2, k_1)$	$\lambda \cdot \min(\ x - y\ ^2, k_2)$	k_1, λ, k_2	$[0.1, 1] \times [0.1, 1] \times [0.1, 1]$	$10 \times 10 \times 10$
Andrew-Andrew	$1 - e^{-k_1 x^2}$	$\lambda \cdot \left(1 - e^{-k_2(x-y)^2}\right)$	k_1, λ, k_2	$[0.1, 1] \times [0.1, 1] \times [0.1, 1]$	$10 \times 10 \times 10$

Al ser todos estos, método iterativos, es necesario fijar un número máximo de iteraciones en caso de que no haya convergencia. Una segmentación típica para Graph-cut requiere alrededor de 4 iteraciones. Para dejar un margen amplio, fijamos 40 iteraciones en la etapa de exploración del espacio de parámetros. Para la etapa de validación el número de iteraciones se incrementará a 100. Sin embargo es importante mencionar que en todas las pruebas los algoritmos basados en Graph-cuts convergieron antes de llegar al límite de iteraciones. En el caso de HMMF cada iteración es mucho menos costosa que una iteración de Graph-cut. Una segmentación típica para este métodos requiere de alrededor de 50 iteraciones. Para la exploración del espacio de parámetros se utilizaron 200 iteraciones. Para la etapa de validación el número de iteraciones se incrementará a 1000 (para EC-QMMF también usaremos 1000 iteraciones en la etapa de validación).

La medida de error que utilizaremos para la comparación cuantitativa será el error cuadrático medio (MSE por *Mean Squared Error*), promediado sobre todos los pixeles de la imagen

$$MSE(\hat{f}) = \frac{1}{N} \sum_{r \in \mathcal{S}} (\hat{f}_r - f_r)^2$$

donde f es la imagen verdadera, \hat{f} es la *imagen estimada* por alguno de los métodos y N es el número de pixeles de la imagen.

En este caso los modelos que utilizamos son constantes por lo que la imagen estimada es simplemente

$$\hat{f}_r = \mu_{\hat{c}_r}$$

donde \hat{c}_r es la clase a la cual fue asignado el pixel r por el método en cuestión y μ_{q_k} es el nivel de gris estimado para la clase k .

Para cada juego de parámetros, se ejecutó el algoritmo correspondiente con 11 imágenes de entrada, cada una con diferente nivel de ruido (etapa de exploración), los parámetros *óptimos* se eligieron de tal

forma que al dejarlos fijos, se minimiza el MSE (promediado sobre los 11 niveles de ruido).

A continuación generamos 10 imágenes por cada nivel de ruido (en total 110 nuevas imágenes) y evaluamos el desempeño de cada método (etapa de validación). De este modo, para cada nivel de ruido y para cada método tenemos un conjunto de 10 imágenes estimadas correspondientes $\{\hat{f}_r^{(1)}, \hat{f}_r^{(2)}, \dots, \hat{f}_r^{(10)}\}$. En la figura 6.1 se muestra el promedio del MSE (\overline{MSE}) para cada nivel de ruido, dado por

$$\overline{MSE} = \frac{1}{10} \sum_{i=1}^{10} MSE(\hat{f}_r^{(i)})$$

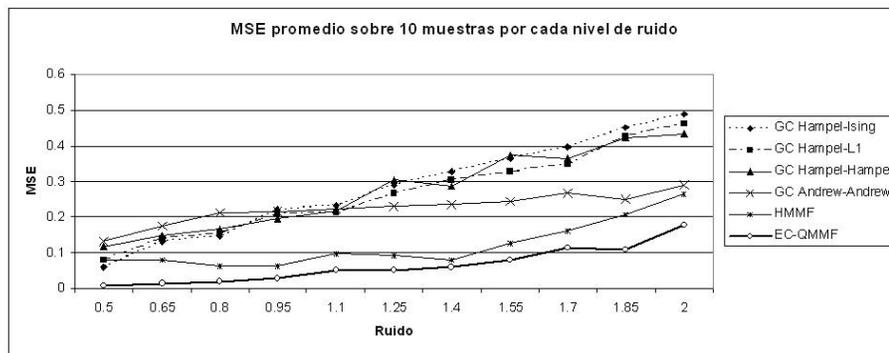


Figura 6.1: Comparación del error cuadrático medio obtenido para diferentes niveles de ruido. Cada punto corresponde al promedio del error sobre 10 imágenes con el mismo nivel de ruido (para su mejor identificación las etiquetas están ordenadas de acuerdo al valor del último nivel de ruido).

En la figura 6.2 ilustramos la variabilidad del error del estimador MAP para cada nivel de ruido, obtenido con las cuatro funciones de energía, podemos observar que la variabilidad del error de los estimadores obtenidos con el potencial de Andrew es mucho menor que el resto.

En la figura 6.3 ilustramos la variabilidad del error de los estimadores obtenidos usando HMMF y EC-QMMF, además de presentar una media de error muy por debajo del resto de los métodos, vemos que la variabilidad del error de EC-QMMF es muy pequeña lo cual muestra la consistencia de los resultados. Solamente se observa una variabilidad alta para niveles extremos de ruido, muy raramente vistos en la práctica.

En la siguiente tabla se muestra el conjunto de parámetros óptimos encontrados para cada función de energía y la media y la desviación estándar de los “MSE promedios”.

Parámetros óptimos y resultados para las funciones de energía del estudio comparativo			
Energía	Parámetros óptimos	MSE promedio	Desv. Est. del MSE promedio
Hampel-Ising	$k_1 = 0.9526, \lambda = 0.4789$	0.2828	0.1370
Hampel- L_1	$k_1 = 1.0, \lambda = 0.6, k_2 = 0.9$	0.267	0.1202
Hampel-Hampel	$k_1 = 0.8, \lambda = 0.6, k_2 = 0.8$	0.2757	0.1131
Andrew-Andrew	$k_1 = 0.1, \lambda = 0.5, k_2 = 0.5$	0.2246	0.0432
HMMF	$\lambda = 0.65, \sigma = 3.475$	0.1192	0.0652
EC-QMMF	$\lambda = 4, \mu = 0.5$	0.0652	0.0519

En el caso de EC-QMMF, para fijar el valor de σ utilizamos el método de estimación de por ventanas descrito al inicio de este capítulo.

En la figura 6.4 se muestran ejemplos de las imágenes utilizadas para la comparación y los mejores resultados obtenidos con los métodos bajo comparación. En esta figura, se eligió para cada método, la mejor segmentación obtenida de entre todos los juegos de parámetros para la imagen que se usó en el proceso de exploración (esta imagen es diferente de las 10 que se usaron en la etapa de validación). En el caso de EC-QMMF fue necesario variar un poco los parámetros (manualmente) en el último nivel de ruido. El objetivo de esta imagen es mostrar los mejores resultados que se pueden lograr con cada método ajustando adecuadamente los parámetros.

En la imagen 6.5 se muestra para cada método y cada nivel de ruido, la mejor segmentación de entre las 10 imágenes de prueba correspondientes de la etapa de validación, pero esta vez dejando fijos los parámetros antes reportados. Obsérvese que la segmentación obtenida con EC-QMMF es incluso mejor que la que se obtuvo en la figura 6.4 aún cuando en aquel caso se eligieron mejores parámetros para esa imagen en particular, esto simplemente significa que la elección de los parámetros que sugerimos antes, y que dejamos fijos en la etapa de validación, es adecuada y el problema fue que la imagen de la etapa de exploración fue de las más problemáticas para EC-QMMF.

La figura 6.6 muestra una comparación del tiempo de ejecución del método propuesto contra la modificación original de HMMF y los dos algoritmos basados en Graph-cuts, el algoritmo α -Expansion se probó con los potenciales de Hampel en el término de datos y la norma L_1 en el término de regularización, el algoritmo $\alpha - \beta$ Swap se probó con los potenciales de Andrew. Se ejecutaron los cuatro algoritmos hasta convergencia sobre la imagen sintética antes mostrada con ruido gaussiano con $\sigma = 0.5$ ya que típicamente en las aplicaciones interesantes no se encuentran grandes niveles de ruido. La gráfica

muestra el tiempo de ejecución en centésimas de segundo para la misma imagen en función del número de modelos. Estos tiempos fueron obtenidos usando un procesador AMD-Athlon XP 3200+, a 2 GHz, con 1GB de memoria. Podemos observar que el método propuesto EC-QMMF es significativamente más eficiente que el resto, en particular es notable la diferencia con su contraparte HMMF, esto se debe a lo relativamente compleja que resulta la función de energía de HMMF y la forma como se manejan las restricciones. Por su parte la función de energía de EC-QMMF es cuadrática y las restricciones se manejan muy eficientemente. El criterio de convergencia para los métodos basados en grafos son los que se indican en [5], el criterio de convergencia de HMMF y EC-QMMF es el mismo para ambos (tal como describimos antes) fijando una tolerancia $\epsilon = \sqrt{10^{-5}}$.

Una ventaja adicional es el control sobre la entropía del campo de probabilidades estimado. Podemos obtener diferentes estimaciones simplemente modificando el parámetro μ . La figura 6.7 muestra diferentes estimaciones, en cada imagen se grafica, para cada sitio r el valor máximo de $p(r)$. Claramente la mayor incertidumbre se encuentra alrededor de los bordes de la imagen observada.

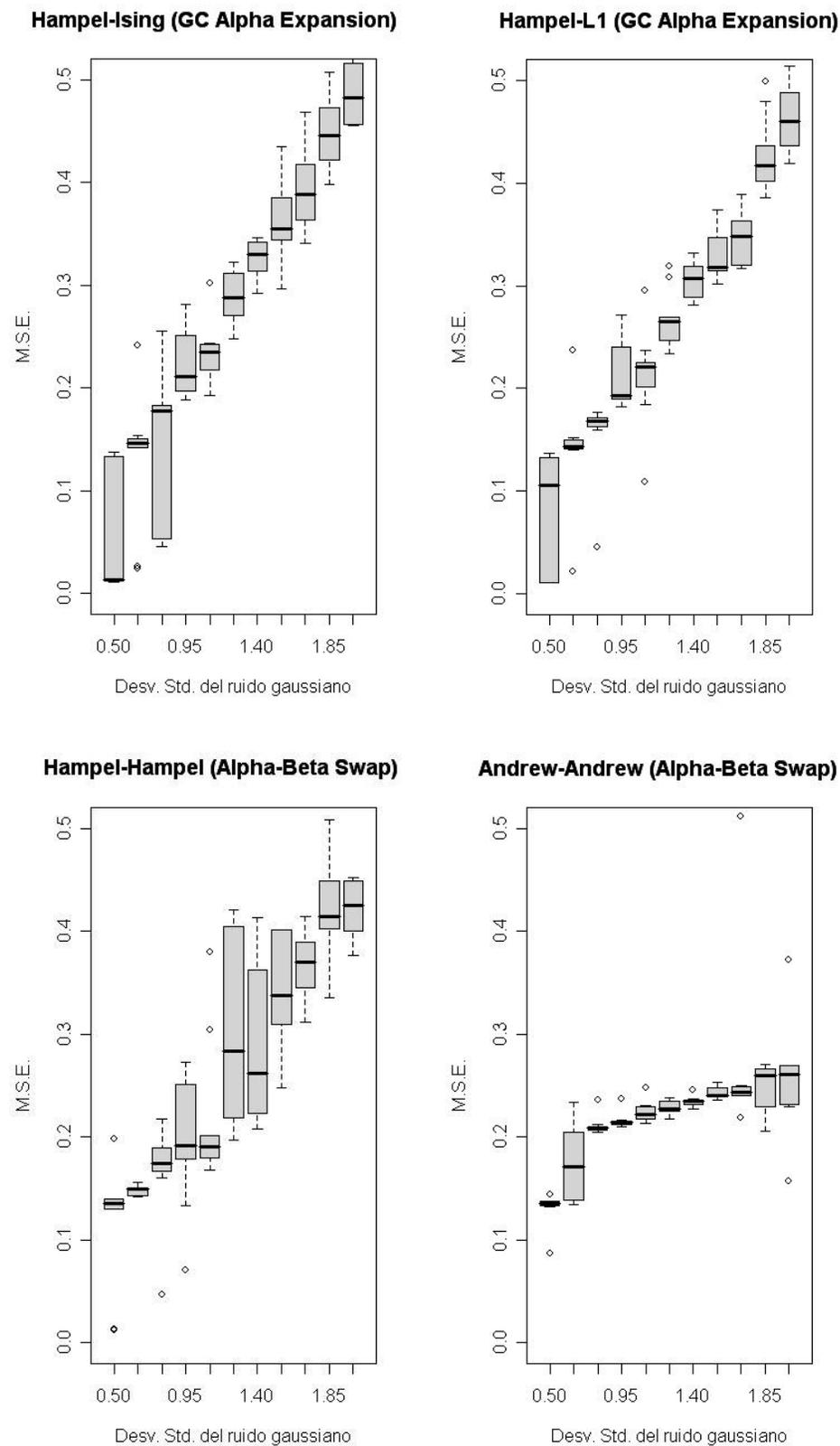


Figura 6.2: Variabilidad del error cuadrático medio de los estimadores MAP usando diferentes funciones de energía. Cada Box-Plot corresponde al error obtenido en 10 imágenes con el mismo nivel de ruido (ver texto).

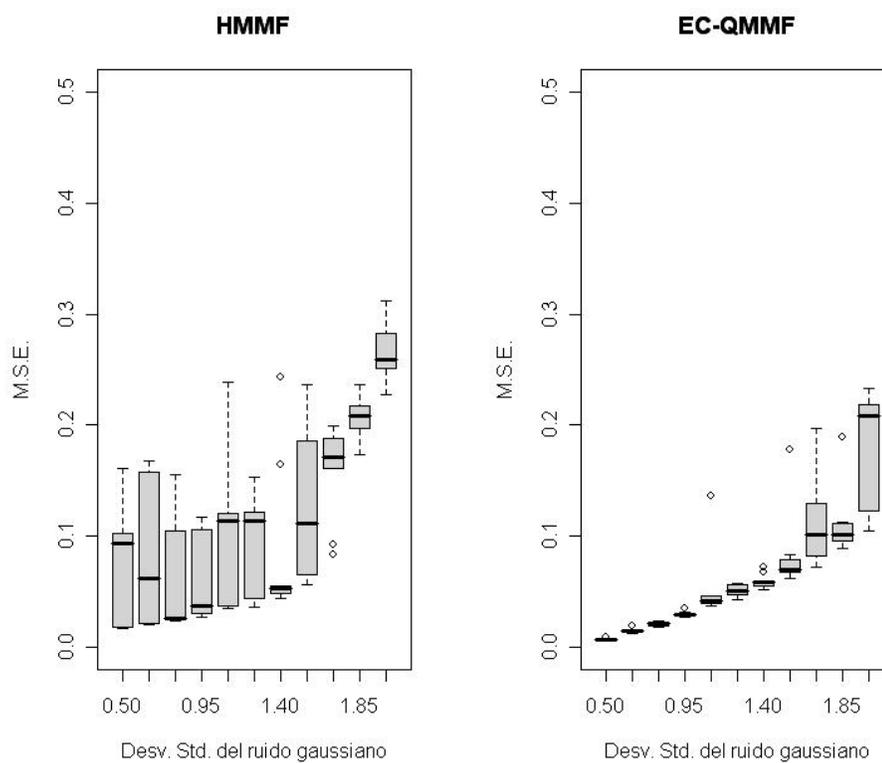


Figura 6.3: Variabilidad del error cuadrático medio de los estimadores obtenidos mediante HMMF y EC-QMMF. Cada Box-Plot corresponde al error obtenido en 10 imágenes con el mismo nivel de ruido (ver texto).

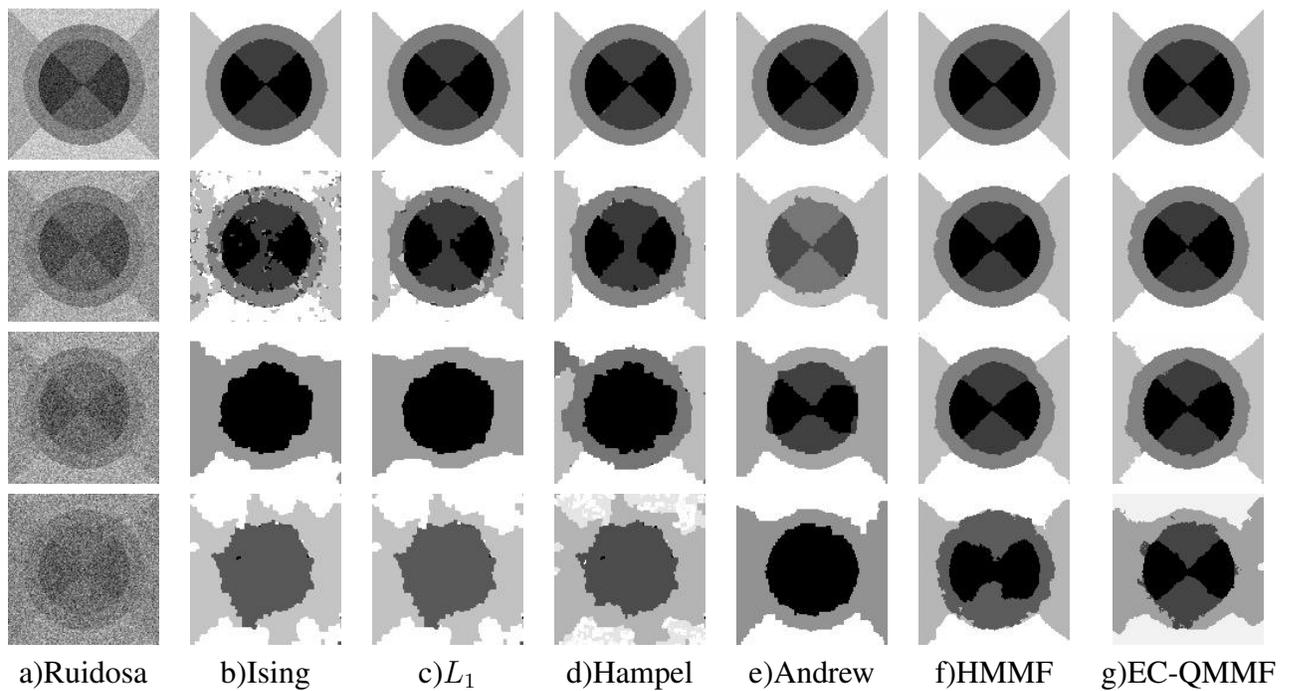


Figura 6.4: Resultados cualitativos usando imágenes sintéticas con ruido gaussiano con $\sigma = 0.5$ (fila 1), 0.95(fila 2), 1.40(fila 3), 2(fila 4). Los estimadores utilizados fueron el estimador MAP para diferentes potenciales (columnas b)-e) y los obtenidos con los modelos HMMF(columna f)) y EC-QMMF(columna g)).

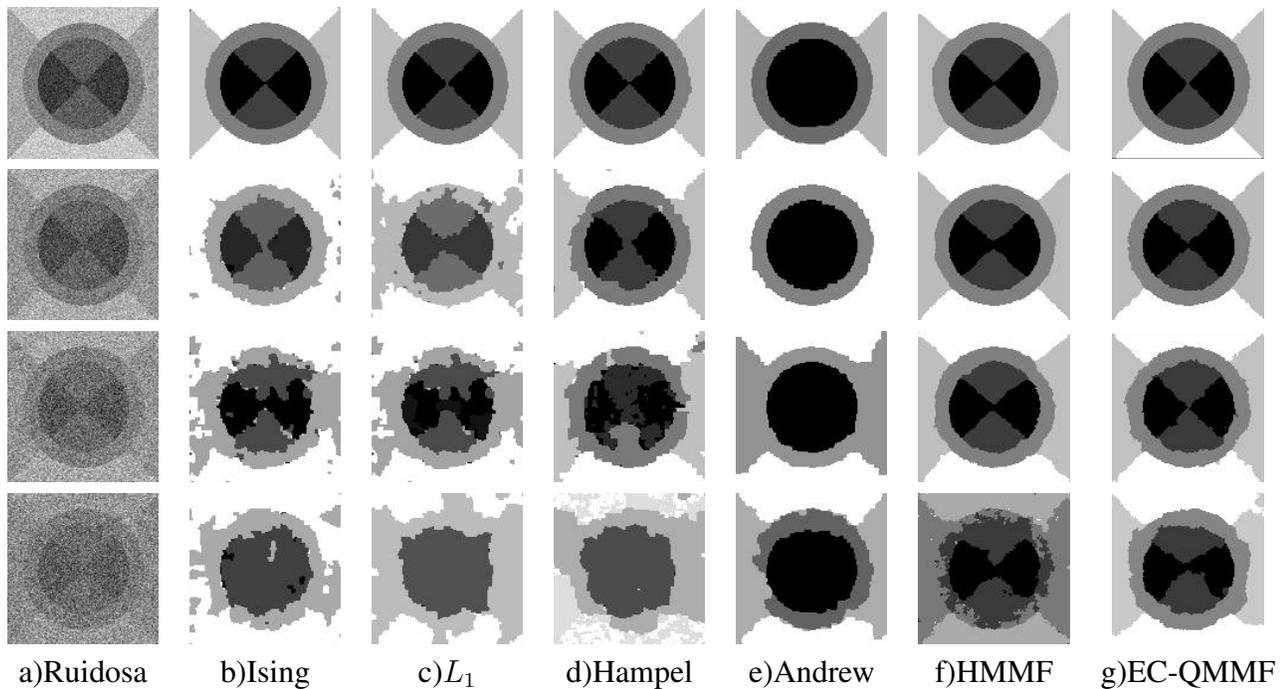


Figura 6.5: Resultados cualitativos usando imágenes sintéticas con ruido gaussiano con $\sigma = 0.5$ (fila 1), 0.95 (fila 2), 1.40 (fila 3), 2 (fila 4). Los estimadores utilizados fueron el estimador MAP para diferentes potenciales (columnas b-e)) y los obtenidos con los modelos HMMF(columna f)) y EC-QMMF(columna g)).

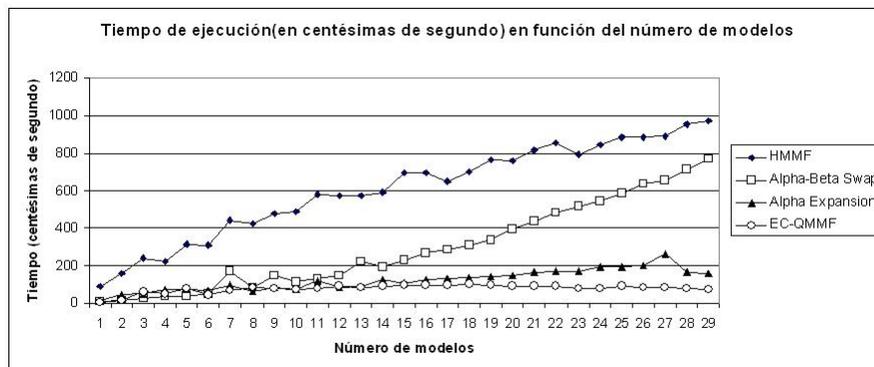


Figura 6.6: Tiempo de ejecución en función del número de modelos (constantes).

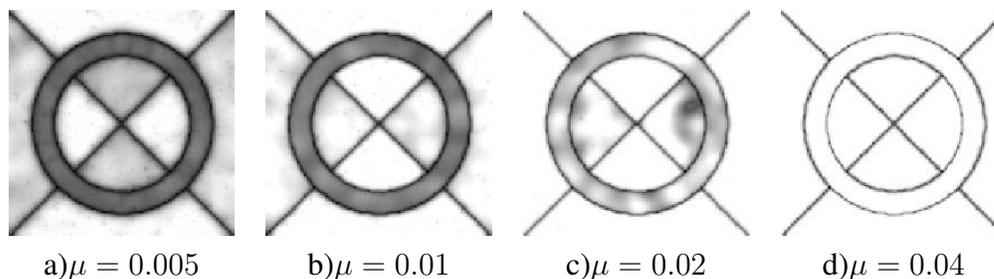


Figura 6.7: Ilustración de la entropía del campo de medida estimado para diferentes valores de μ .

Capítulo 7

Aplicaciones

*The science of today
is the technology of tomorrow.*

Edward Teller.

Una aplicación de gran interés es la segmentación de imágenes de resonancia magnética del cerebro en función del tipo de tejido. La tarea es clasificar los voxeles en tres clases: materia blanca (MB), materia gris (MG) y líquido cefalo-raquídeo (LCR). Diversas técnicas han sido utilizadas para atacar este problema desde métodos basados en umbrales[35] y técnicas tradicionales de agrupamiento (*clustering*) [9] [21] hasta métodos de segmentación paramétrica que consideran las interacciones espaciales de los voxeles y están basados en modelación mediante Campos Aleatorios Markovianos y estimación Bayesiana [16][24]. Debido a la naturaleza de las imágenes, en las cuales además del problema de ruido, los niveles de gris que caracterizan a cada tipo de tejido no son constantes (o casi constantes) sino que pueden variar a lo largo de la imagen, los métodos más exitosos han sido éstos últimos.

Mostraremos el desempeño de EC-QMMF usando tanto imágenes simuladas como reales. La característica que define la segmentación en este caso es la intensidad (nivel de gris) de los voxeles. Como vimos en la sección anterior EC-QMMF es muy robusto, por lo que solamente debemos ocuparnos por modelar adecuadamente el efecto de las inhomogeneidades. El modelo que usaremos será el de splines de producto tensorial.

7.1. Splines de producto tensorial

El modelo que usaremos para la intensidad de cada tipo de tejido está definido por las funciones

$$\phi_k(r) = \sum_{j=1}^M \theta_{k,j} N_j(r)$$

donde

$$N_j(x, y) = N_0(x - x_j, y - y_j), \quad N_0(x, y) = B(x/\delta)B(y/\delta)$$

son traslaciones de N_0 hacia un conjunto de *puntos de control* que forman una retícula sobre la región de interés (figura 7.1). La separación entre los puntos de control δ define la escala a la que utilizaremos la función base B definida por

$$B(x) = \begin{cases} \frac{(1.5-2x^2)}{2} & |x| \in [0, 0.5] \\ \frac{(x^2-3|x|+2.25)}{2} & |x| \in [0.5, 1.5] \\ 0 & |x| \geq 1.5 \end{cases}$$

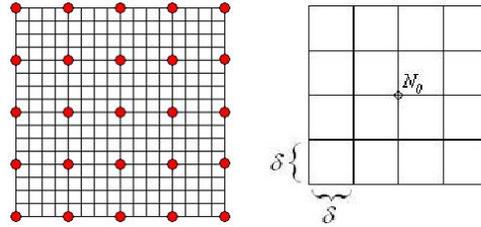


Figura 7.1: Puntos de control en el modelo de splines de producto tensorial.

Notemos que el paso de optimización respecto a p no varía con el uso de splines ya que solamente nos interesa $\log(v_k(r))$. Lo único que varía es el cálculo de los parámetros óptimos de las splines dado el valor de p . La energía está dada por

$$U(p, \theta) = \sum_{r \in S} \sum_k \frac{1}{2\sigma^2} ((\phi_k(r) - g(r))^2 - \mu) p_k(r)^2 + \lambda \sum_{\langle r, s \rangle \in C_2} \sum_k (p_k(r) - p_k(s))^2$$

Experimentalmente este modelo presenta problemas de inestabilidad debido a que los valores de $\theta_{k,j}$ no están ligados entre sí. Este problema puede ser resuelto regularizando los parámetros, es decir modelamos los parámetros $\theta_{k,j}$ como un campo aleatorio Markoviano, sobre el cual introducimos información a priori

$$P(\theta) = \frac{1}{Z_\theta} \exp^{-\eta \sum_{\langle i, j \rangle} (\theta_i - \theta_j)^2}$$

donde $\langle i, j \rangle$ indica que la suma se toma sobre todos los pares (i, j) de puntos de control vecinos entre si. Agregando este término a la función de energía obtenemos

$$U(p, \theta) = \sum_{r \in \mathcal{S}} \sum_k \frac{1}{2\sigma^2} ((\phi_k(r) - g(r))^2 - \mu) p_k(r)^2 + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} \sum_k (p_k(r) - p_k(s))^2 + \eta \sum_{\langle i, j \rangle} (\theta_i - \theta_j)^2$$

Para obtener los parámetros óptimos:

$$\frac{\partial U}{\partial \theta_{k,j}}(p, \theta) = 2 \sum_{r \in \mathcal{S}} \frac{\partial \phi_k}{\partial \theta_{k,j}}(r) (\phi_k(r) - g(r)) p_k(r)^2 + 2\eta \sum_{i \in \mathcal{N}_j} (\theta_{k,j} - \theta_{k,i})$$

donde \mathcal{N}_j es el conjunto de puntos de control vecinos del punto de control j . Además

$$\frac{\partial \phi_k}{\partial \theta_{k,j}}(r) = N_j(r)$$

entonces para los parámetros óptimos θ ,

$$\eta \theta_{k,j} \#\mathcal{N}_j + \sum_{r \in \mathcal{S}} N_j(r) \sum_i \theta_{k,i} N_i(r) p_k(r)^2 = \eta \sum_{i \in \mathcal{N}_j} \theta_{k,i} + \sum_{r \in \mathcal{S}} N_j(r) g(r) p_k(r)^2$$

Este es un sistema de ecuaciones lineales que podemos resolver usando cualquier método estándar como eliminación gaussiana o descomposición LU, o bien podemos aproximar la solución usando métodos iterativos tales como Jacobi o Gauss-Seidel [8].

El paso de actualización de Jacobi es:

$$\theta_{k,j}^{(t+1)} = \frac{\sum_{r \in \mathcal{S}} N_j(r) g(r) p_k(r)^2 + \eta \sum_{i \in \mathcal{N}_j} \theta_{k,i} - \sum_{r \in \mathcal{S}} N_j(r) \sum_{i \neq j} \theta_{k,i}^{(t)} N_i(r) p_k(r)^2}{\sum_{r \in \mathcal{S}} N_j(r)^2 p_k(r)^2 + \eta \#\mathcal{N}_j}$$

El paso de actualización de Gauss-Seidel es análogo, simplemente utilizamos siempre los últimos valores calculados de $\theta_{k,j}$.

$$\theta_{k,j}^{(t+1)} = \frac{\sum_{r \in \mathcal{S}} N_j(r) g(r) p_k(r)^2 + \eta \sum_{i \in \mathcal{N}_j} \theta_{k,i} - \sum_{r \in \mathcal{S}} N_j(r) \left[\sum_{i < j} \theta_{k,i}^{(t+1)} N_i(r) p_k(r)^2 + \sum_{i > j} \theta_{k,i}^{(t)} N_i(r) p_k(r)^2 \right]}{\sum_{r \in \mathcal{S}} N_j(r)^2 p_k(r)^2 + \eta \#\mathcal{N}_j}$$

En las siguientes secciones se muestran los resultados obtenidos en dos conjuntos de datos muy utilizados en la literatura: el conjunto de imágenes sintéticas de Montreal, las cuales fueron generadas usando el simulador de resonancia magnética *Brain WEB* [10] [11] [24] y el conjunto de imágenes reales de la Escuela de Medicina de Harvard publicados por el Centro de Analysis Morfométricos del Hospital General de Massachusetts [20].

7.2. IRM sintéticas

El conjunto de datos sintéticos fue generado a partir de un modelo anatómico ¹. Para cada voxel de este modelo anatómico se especifica la proporción de cada tejido que está presente en el mismo. A partir de estos datos se realiza una simulación en la que se modelan dos artefactos que podrían estar presentes en las imágenes de resonancia magnética: ruido e inhomogeneidades espaciales. En las siguientes secciones trabajaremos con simulaciones de MRI con 0% y 40% de inhomogeneidades espaciales y diferentes niveles de ruido: 1%, 3%, 5%, 7% y 9%. En la figura 7.2 se muestran ejemplos del tipo de imágenes con las que trabajaremos en esta sección.

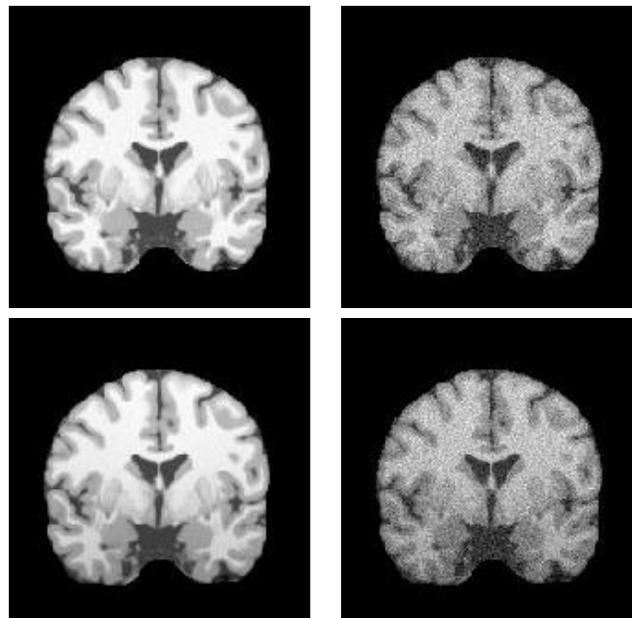


Figura 7.2: Ejemplos de los datos simulados de Montreal. Fila 1: sin inhomogeneidades espaciales. Fila 2: con 40% de inhomogeneidades espaciales. Columna 1: 1% de ruido. Columna 2: 9% de ruido.

Para el conjunto de datos sin inhomogeneidades espaciales basta realizar una segmentación usando modelos constantes. Para las segmentaciones de los datos con inhomogeneidades espaciales se utilizó primero una segmentación usando modelos constantes, luego las splines fueron inicializadas en el valor constante de cada clase obtenido con esta segmentación. Finalmente se utilizó el modelo de splines descrito en la sección anterior para *refinar* los resultados.

¹<http://www.bic.mni.mcgill.ca/brainweb>

Un detalle importante que hay que cuidar es el de hacer corresponder cada clase de la segmentación a cada clase de interés, en este caso, la clase con nivel de gris más bajo corresponde al LCR y la clase con nivel más alto a la materia blanca, etc. Efectuamos esta correspondencia en cada iteración de los modelos **constantes** simplemente ordenándolos por nivel de gris, de esta forma la clase 0 corresponde al LCR, la clase 1 a la MG y la 2 a la MB. Al efectuar el ajuste mediante splines ya no es necesario reordenar los modelos.

El procedimiento descrito en la sección anterior corresponde a **una imagen** (o *corte*) del volumen de datos. Los resultados que mostramos fueron obtenidos segmentando cada corte por separado. Tenemos tres posibilidades para hacer esto, que son considerar cortes coronales, axiales o sagitales, en la figura 7.3 se muestran ejemplos de estos tipos de corte. Las tres opciones son buenas en los cortes intermedios del volumen pero en los cortes extremos puede haber ausencia de alguna de las clases (típicamente en los cortes extremos puede no haber materia blanca) en este caso naturalmente obtenemos una sobre-segmentación (figura 7.4). Para resolver este problema efectuamos dos segmentaciones, una considerando cortes coronales y la otra considerando cortes axiales, luego construimos una segmentación corregida tomando para cada voxel la clasificación del modelo más confiable en el voxel en cuestión. El criterio para decidir cuál segmentación es más confiable simplemente para cada voxel, verificamos qué tan alejado está el voxel en cuestión al centro del volumen respecto a cada tipo de corte. Por ejemplo los cortes coronales corresponden a desplazamientos en la coordenada y , los cortes axiales corresponden a desplazamientos en z , sea $c = (c_x, c_y, c_z)$ el centro del volumen y $r = (x, y, z)$ cualquier voxel. Llamaré *segmentación coronal* de r a la clase asignada a r por la segmentación hecha usando cortes coronales, análogamente usaré el término *segmentación axial*. Decimos que la clasificación coronal de r es *mas confiable* que su clasificación axial si $|c_y - y| < |c_z - z|$ (es decir, r se encuentra más cerca del centro respecto a los cortes coronales). Para obtener la segmentación final, simplemente asignamos a cada voxel su clasificación más confiable, ya sea axial o coronal.

Los parámetros de EC-QMMF se mantuvieron fijos **en todos los experimentos**. Se utilizó el *procedimiento de ventanas* para estimar σ . Los parámetros para la segmentación inicial con modelos constantes fueron $\lambda = 1.4, \mu = 1$. Para la segmentación usando el modelo de splines los parámetros fueron $\lambda = 1, \mu = 0.5, \eta = 5$, el número máximo de iteraciones fue 200 y la tolerancia $\epsilon = 10^{-5}$.

Para este conjunto de datos, en la literatura se reportan los resultados en términos del siguiente índice

$$\xi_k = \frac{2V_{GPk}}{V_{Pk} + V_{Gk}}$$

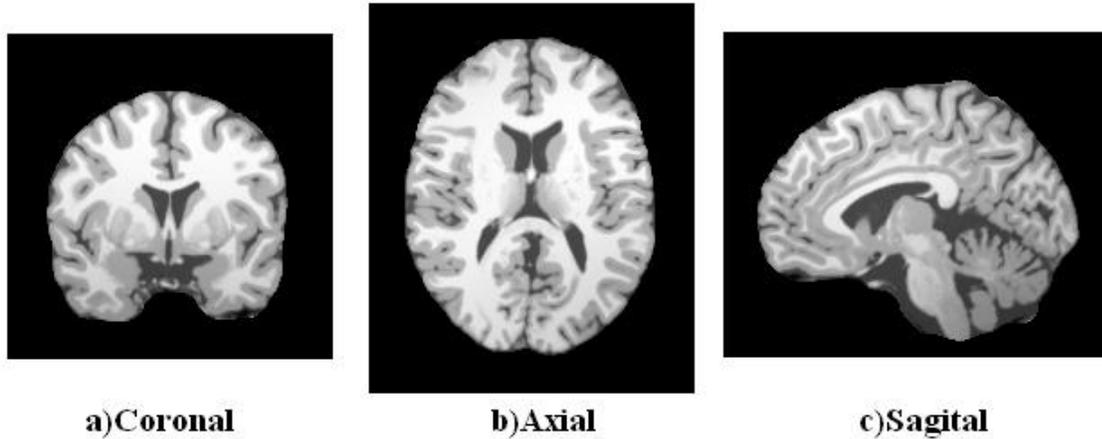


Figura 7.3: Tipos de corte del volumen de datos. Tomados del conjunto de datos sintéticos de Montreal (<http://www.bic.mni.mcgill.ca/brainweb>).

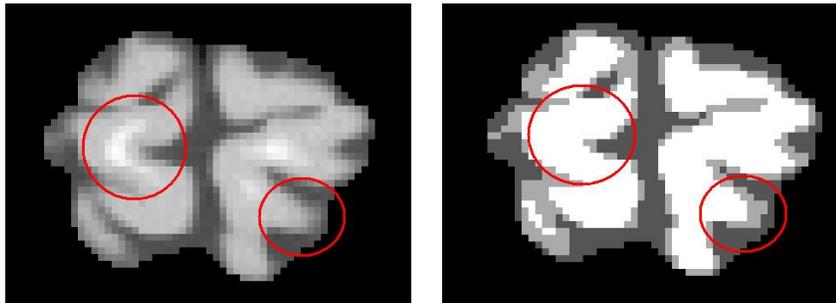


Figura 7.4: En los extremos, en ausencia (o casi ausencia) de materia blanca, obtenemos una sobre-segmentación de la materia gris al mismo tiempo que la clase asociada a la materia blanca desaparece.

donde V_{GPk} es el número de voxeles correctamente asignados a la clase k , V_{Pk} es el número de voxeles asignados a la clase k (ya sea correcta o incorrectamente) y V_{Gk} es el número de voxeles que realmente pertenecen a la clase k (para ésto se utiliza el modelo anatómico). Claramente para una segmentación *perfecta*, $\xi_k = 1$ para todos los modelos k . El valor de este índice será menor conforme haya más voxeles mal clasificados, hasta llegar a cero.

La última consideración que necesitamos hacer tiene que ver con la selección del conjunto de datos de interés. Los voxeles fuera del cerebro pueden tratarse de dos maneras distintas: introducir un modelo adicional y hacer que el método de segmentación se encargue de separarlo o bien restringir el procedimiento a los voxeles del cerebro (dado que contamos con un modelo anatómico, podemos descartar los voxeles del fondo).

En las siguientes dos secciones se muestran los resultados obtenidos usando cada uno de estos procedimientos.

7.2.1. Introduciendo un modelo adicional

Este método es muy simple, basta realizar la segmentación a partir de 4 modelos en lugar de 3. En la figura 7.5 se muestra la gráfica de desempeño obtenido para el modelo EC-QMMF y HMMF.

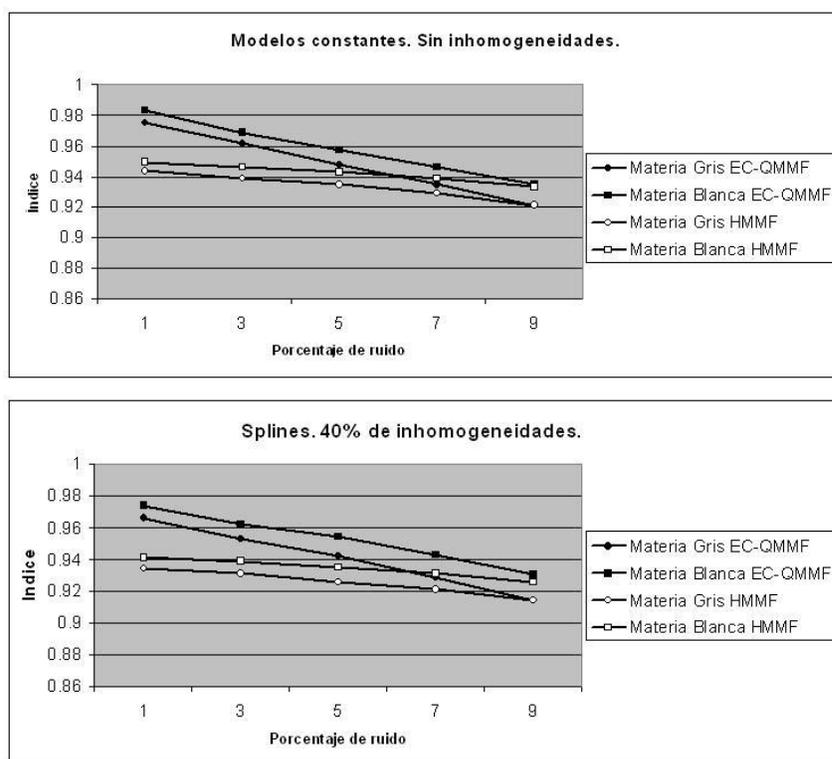


Figura 7.5: Resultados de la segmentación de imágenes de resonancia magnética del cerebro (simulaciones de Montreal). Segmentación usando 4 modelos sobre las imágenes originales.

Aunque no mostramos el índice de clasificación para el fondo, prácticamente todos sus voxels son bien segmentados.

7.2.2. Considerando únicamente la región del cerebro

Estamos suponiendo que contamos con un modelo anatómico a partir del cual podemos etiquetar los voxels como cerebro y no-cerebro. Los resultados que mostramos en esta sección fueron obtenidos con

el procedimiento descrito en las secciones anteriores tomando como región de interés el subconjunto de sitios $\hat{S} \subset \mathcal{S}$ que pertenecen al cerebro.

En la figura 7.6 se muestran las gráficas de desempeño.

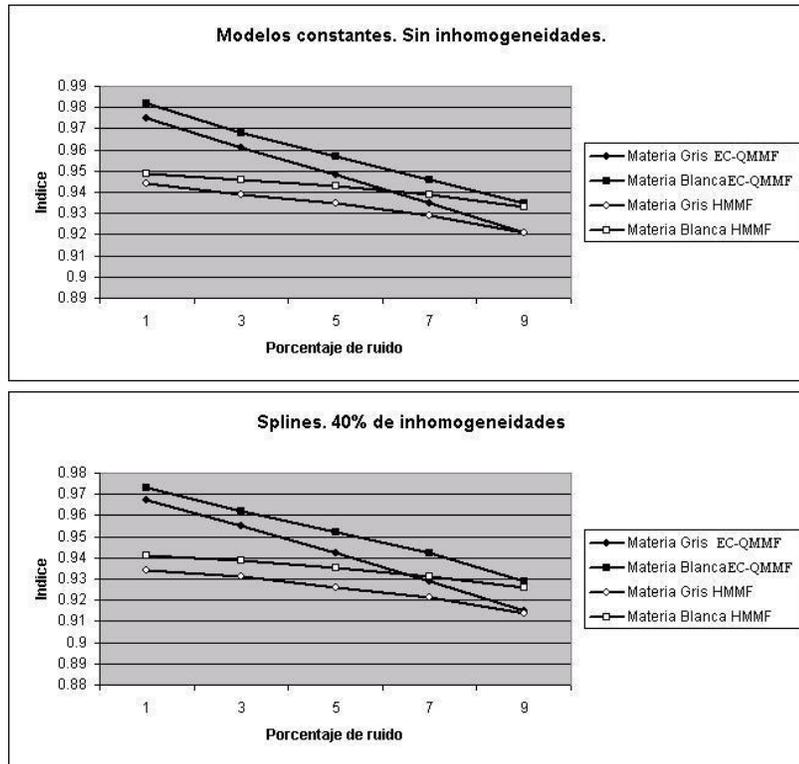


Figura 7.6: Resultados de la segmentación de imágenes de resonancia magnética del cerebro (simulaciones de Montreal). Segmentación usando 3 modelos considerando únicamente la región del cerebro.

Los resultados son prácticamente los mismos, esto se debe precisamente a que el fondo fue muy bien segmentado en el procedimiento de la subsección anterior.

7.3. IRM reales (conjunto de datos de Harvard)

En el conjunto de datos de Harvard, la discretización es diferente de la anterior, cada voxel corresponde a un volumen de $1 \times 1 \times 3 \text{ mm}^3$. (voxels mas grandes), esto ocasiona que en un mismo voxel haya mezclas de más de un tejido, este fenómeno es conocido como *efecto de volumen parcial*. Debido a esto existen en los datos de Harvard algunos voxels cuya composición es una mezcla de LCR y MG, si

observamos la segmentación manual (Figura 7.10), nos damos cuenta de que esta mezcla fue asignada por el experto a la clase MG. Por tanto para realizar esta segmentación es necesario introducir una cuarta clase que corresponde a los voxeles de mezcla, es decir, haremos la segmentación usando 4 modelos.

Otra importante diferencia del método utilizado en este conjunto de datos respecto al que utilizamos con el conjunto de Montreal discutido en la sección anterior es que no es conveniente combinar el resultado de las segmentaciones sobre los cortes de menor resolución. En la figura 7.7 se muestra el tipo de imágenes considerando los tres tipos de corte. Las segmentaciones sobre los cortes axial y sagital son muy malas, por lo que no es conveniente mezclar los resultados como lo hicimos con los datos de Montreal.

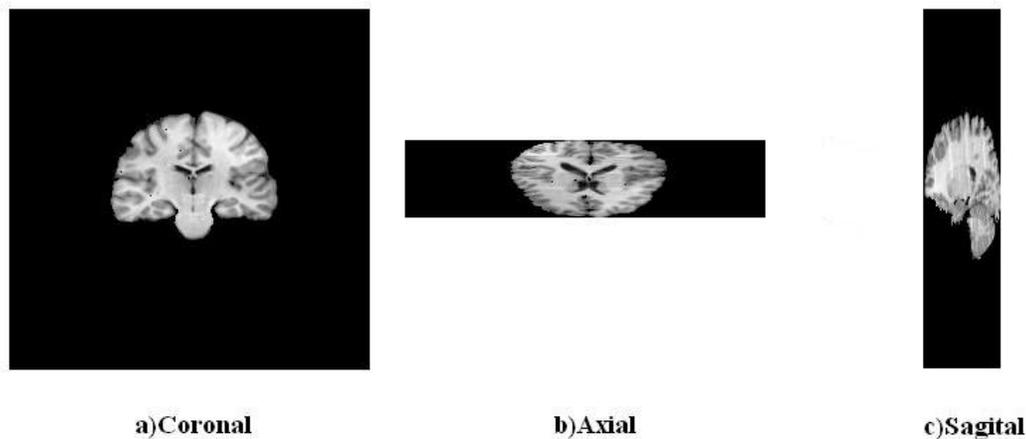


Figura 7.7: Ejemplo de corte coronal, axial y sagital para el conjunto de datos de Harvard(ver texto).

Para este conjunto de datos, los resultados en la literatura se reportan usando el *índice de Tanimoto* definido por:

$$\xi_k = \frac{V_{p_k \cap g_k}}{V_{p_k \cup g_k}}$$

aquí, p_k representa el conjunto de voxeles clasificados como clase k por el método en cuestión (ya sea correcta o incorrectamente), g_k es el conjunto de voxeles que realmente pertenecen a la clase k . Luego $V_{p_k \cap g_k}$ es el número de voxeles en la intersección de los conjuntos p_k y g_k , análogamente $V_{p_k \cup g_k}$.

Compararemos nuestro método contra los resultados reportados para 8 métodos automáticos (ver los detalles de estos métodos en [32]) y el promedio de la segmentación manual de 4 cerebros promediados sobre 2 expertos (este resultado está reportado en [27]).

En este caso tenemos también dos maneras de proceder con respecto al fondo: usar los datos completos o bien considerar solamente la región del cerebro. En las siguientes dos subsecciones se muestran los resultados obtenidos al usar cada uno de estos procedimientos.

7.3.1. Usando solamente los voxeles del cerebro

Al considerar únicamente la región del cerebro, es necesario segmentar usando 4 modelos debido al efecto de volumen parcial que ya se describió antes (2 modelos para la materia gris, uno para la materia blanca y uno mas para el LCR).

El problema con esta opción es que en los cortes alejados del centro del volumen se puede ver que prácticamente el LCR *puro* ha desaparecido (ver figura 7.8) y solamente hay mezcla de LCR y materia gris, esto provocará que haya una sobre-segmentación en alguna de las clases (notar que si consideramos los datos completos este fenómeno desaparece ya que la intensidad el fondo actuará como la clase que corresponde al LCR así que el modelo no desaparecerá).

En la figura 7.9 se muestra el índice de Tanimoto de la materia gris y materia blanca obtenido con EC-QMMF, comparado con los reportados en la literatura.

Los resultados muestran que es necesario hacer algo para corregir la sobresegmentación en los cortes alejados del centro del volumen.

7.3.2. Usando los datos completos

La otra forma de proceder es, como se hizo con los datos de Montreal es usar los datos sin descartar el fondo. Como se ve por ejemplo en la figura 7.7, el nivel de gris del fondo es muy parecido al del LCR (véase por ejemplo la región de los ventrículos, en la parte central del corte coronal) por lo que podríamos en principio segmentar con 4 clases. De estas cuatro clases dos de ellas serán asignadas a la materia gris (por el efecto de volumen parcial), una a la unión fondo - LCR (porque el nivel de gris de estas dos clases es prácticamente el mismo) y la última a la materia blanca.

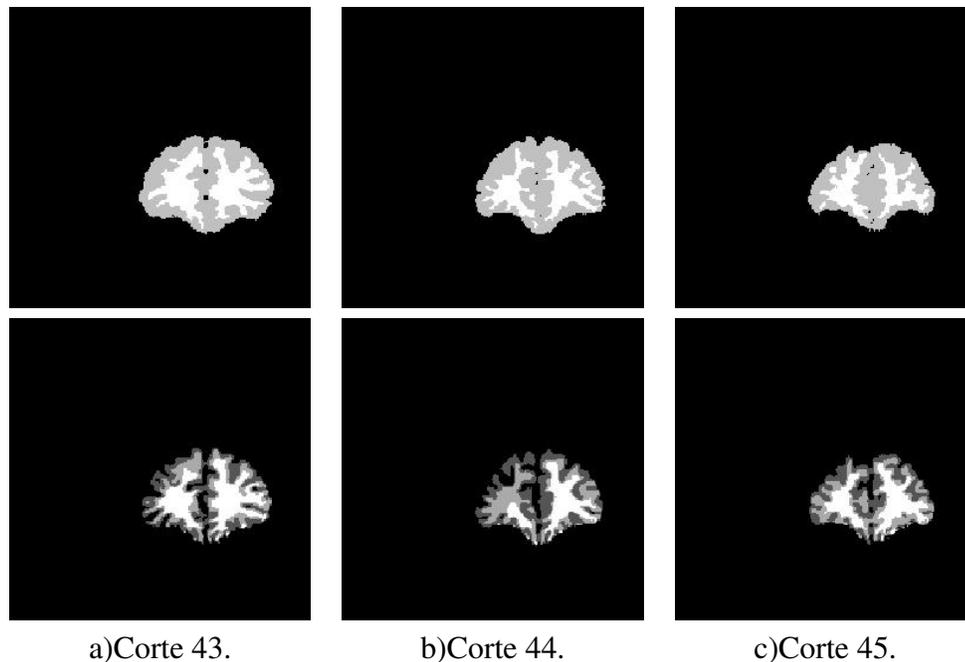


Figura 7.8: Segmentación manual (fila 1) de los cortes coronales extremos del volumen “1_24” y segmentación obtenida usando EC-QMMF (fila 2). Se aprecia la ausencia de LCR y el efecto que causa en la segmentación.

La segmentación final asigna la clase 0 a la unión del fondo y el LCR de los ventrículos. Para hacer esta asignación simplemente tomamos un voxel r que fue clasificado como “clase 0”, luego si en el modelo anatómico r está dentro de la región del cerebro entonces asignamos a r a la clase LCR, si r no está en la región del cerebro entonces lo asignamos al fondo. Las clases 1 y 2 se asignan a la Materia Gris y la clase 3 a la materia blanca.

En la figura 7.11 se muestran las gráficas del índice de Tanimoto comparando el desempeño de HMMF y EC-QMMF.

En la siguiente tabla se muestran la media y la desviación estándar del índice de Tanimoto para diferentes métodos reportados en la literatura.

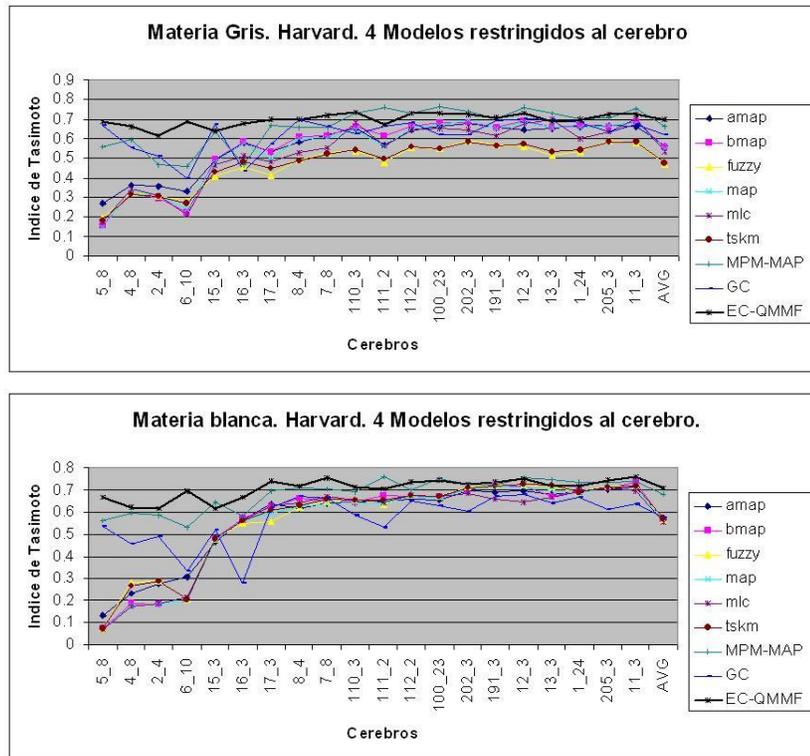


Figura 7.9: Resultados de la segmentación de las imágenes de resonancia magnética del cerebro (Muestras reales de Harvard), utilizando únicamente los voxeles del cerebro.

Resultados numéricos reportados en la literatura y resultados usando EC-QMMF				
	Materia gris		Materia blanca	
	Promedio	Desv. std.	Promedio	Desv. std.
Adaptive MAP	0.564	0.130	0.567	0.180
Biased MAP	0.558	0.169	0.562	0.214
Fuzzy c-means	0.473	0.115	0.567	0.196
Maximum a Posteriori Probability	0.55	0.164	0.554	0.213
Maximum Likelihood	0.535	0.160	0.551	0.207
Tree-Structured k-means	0.477	0.118	0.571	0.198
MPM-MAP	0.662	0.102	0.683	0.072
Graph-cut	0.623	0.087	0.574	0.112
Manual(4 cerebros promediados sobre 2 expertos)	0.876	-	0.832	-
EC-QMMF	0.819	0.028	0.742	0.039

Recientemente se publicó un segundo conjunto de datos de Harvard que consta de 18 nuevos cerebros. Debido a lo reciente de esta publicación no existen en la literatura resultados con los cuales comparar

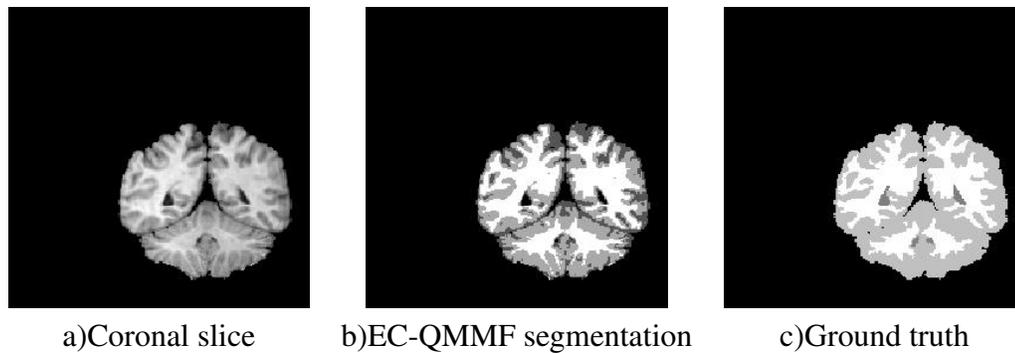


Figura 7.10: Efecto de volumen parcial y comportamiento de EC-QMMF.

nuestro método. La siguiente tabla muestra los resultados numéricos (media y desviación estándar para la materia gris y la materia blanca). La figura 7.12 muestra la gráfica con los resultados para las 18 muestras.

Resultados obtenidos para el nuevo conjunto de datos de Harvard		
	Media	Desv. std.
Materia gris	0.8617	0.0230
Materia blanca	0.7882	0.0283

7.3.3. Tiempo de ejecución

Para establecer un punto de comparación en cuanto al tiempo requerido por los métodos de segmentación, en seguida doy a grandes rasgos los tiempos requeridos por nuestro método en los experimentos con imágenes reales y sintéticas. El equipo utilizado es un procesador Athlon XP 3200+ a 2GHz. (para este tipo de procesadores el rendimiento es un poco mayor) con 1GB de memoria.

Para el conjunto de datos sintéticos de Montreal **sin inhomogeneidades** espaciales (en el que no es necesario el ajuste mediante splines), la segmentación de cada corte coronal requiere alrededor de 0.8 segundos para un total de 217 cortes coronales. por lo que para esta fase del proceso se requiere alrededor de 3 minutos. Para la segmentación de cada corte axial se requiere alrededor de 1 segundo para un total de 181 cortes axiales, el tiempo de ejecución para esta fase del proceso es también de alrededor de 3 minutos. Para segmentar cada volumen se requiere alrededor de 6 minutos.

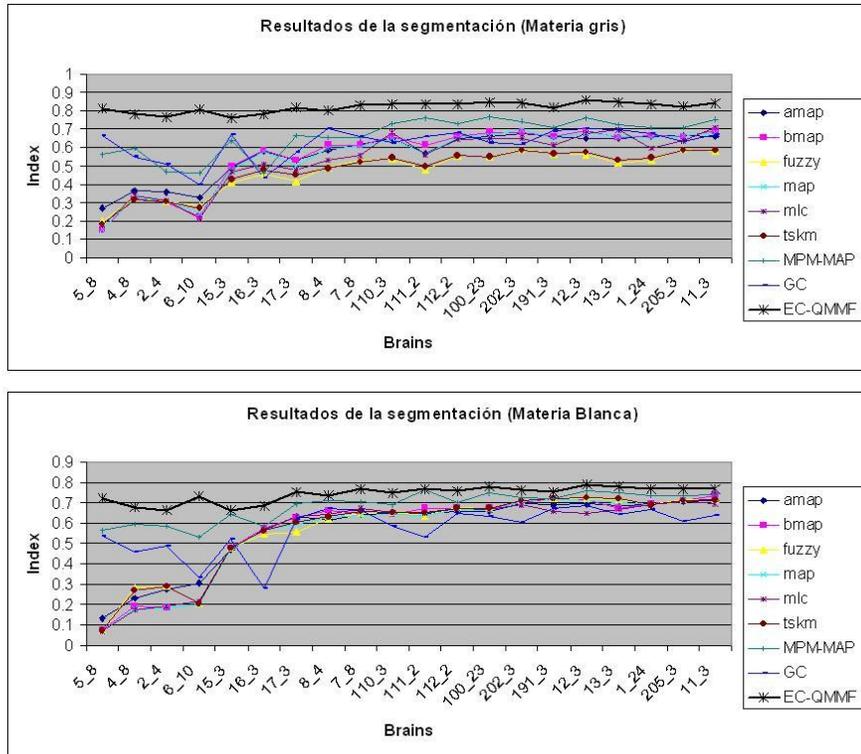


Figura 7.11: Resultados de la segmentación de las imágenes de resonancia magnética del cerebro (conjunto de datos de Harvard), utilizando tanto los voxels del cerebro como los del fondo.

Para el conjunto de datos sintéticos de Montreal **con inhomogeneidades** espaciales (en el que utilizamos el ajuste mediante splines), la segmentación de cada corte coronal requiere alrededor de 65 segundos para un total de 217 cortes coronales. Muchos de ellos requieren mucho menos tiempo, sobre todo en los cortes extremos (estos cortes se segmentan en menos de 1 segundo) por lo que para esta fase del proceso se requiere alrededor de $3\frac{1}{2}$ horas. Para la segmentación de cada corte axial se requiere alrededor de 75 segundos para un total de 181 cortes axiales, un poco más de tiempo pero menos cortes por lo que el tiempo de ejecución para esta fase del proceso es también de alrededor de $3\frac{1}{2}$ horas. Para segmentar cada volumen se requiere alrededor de 7 horas.

Para el conjunto de datos reales de Harvard (en este caso también utilizamos el ajuste mediante splines), la segmentación de cada corte coronal requiere alrededor de 2 minutos, 10 segundos, en este conjunto de prueba el número de cortes varía entre 50 y 60 cortes así que cada cerebro se segmenta en alrededor de 2 horas (recordemos que en este caso solamente se toman las segmentaciones de los cortes coronales).

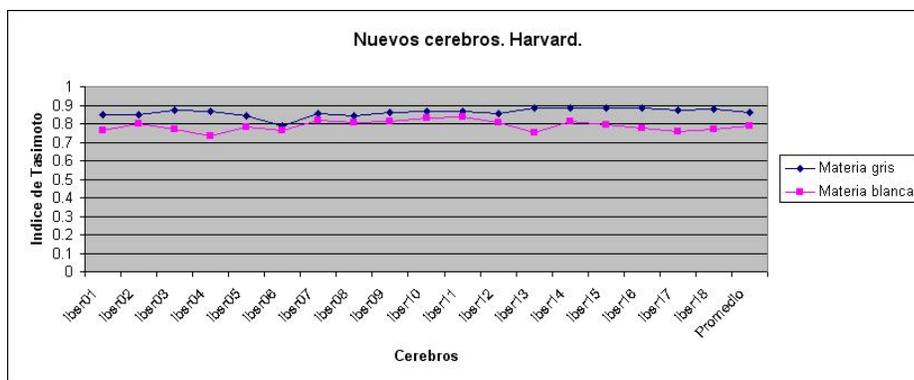


Figura 7.12: Resultados de la segmentación de las imágenes de resonancia magnética del cerebro (18 nuevas muestras reales de Harvard).

7.4. Discusión

Los resultados mostrados en esta primera aplicación son muy superiores a los reportados en la literatura. Entre las ventajas de este método sobre sus predecesores se encuentran

- **Robustés:** EC-QMMF mostró un buen comportamiento aún con niveles relativamente altos de ruido.
- **Eficiencia:** aún cuando en la literatura no se reportan los tiempos de ejecución para la segmentación de un volumen completo, mostramos en el capítulo anterior que este método es mucho más eficiente que su predecesor HMMF y otros métodos aún más costosos computacionalmente como Graph-cut. Incluso, al tratarse de un potencial cuadrático, diversos métodos de optimización pueden ser estudiados para acelerar el proceso, al menos en cuanto al cálculo del campo de medida.
- **Sencillez:** el método iterativo propuesto es muy fácil de implementar.
- **Método libre de parámetros:** en todos los experimentos reportados todos los parámetros se mantuvieron fijos.

Quizá la principal crítica que se le puede hacer a este método es que no es capaz de segmentar correctamente tanto las imágenes sintéticas de Montreal como las imágenes reales de Harvard **usando un único procedimiento**. Sin embargo, al observar cuidadosamente las imágenes se notan demasiadas diferencias, la primera y quizá más importante es la diferencia en la resolución que ya discutimos. Otra diferencia importante tiene que ver con los niveles de gris asociados a cada clase de tejido. En la figura 7.13 se observa que el nivel de gris del fondo es prácticamente el mismo que el del LCR en el conjunto

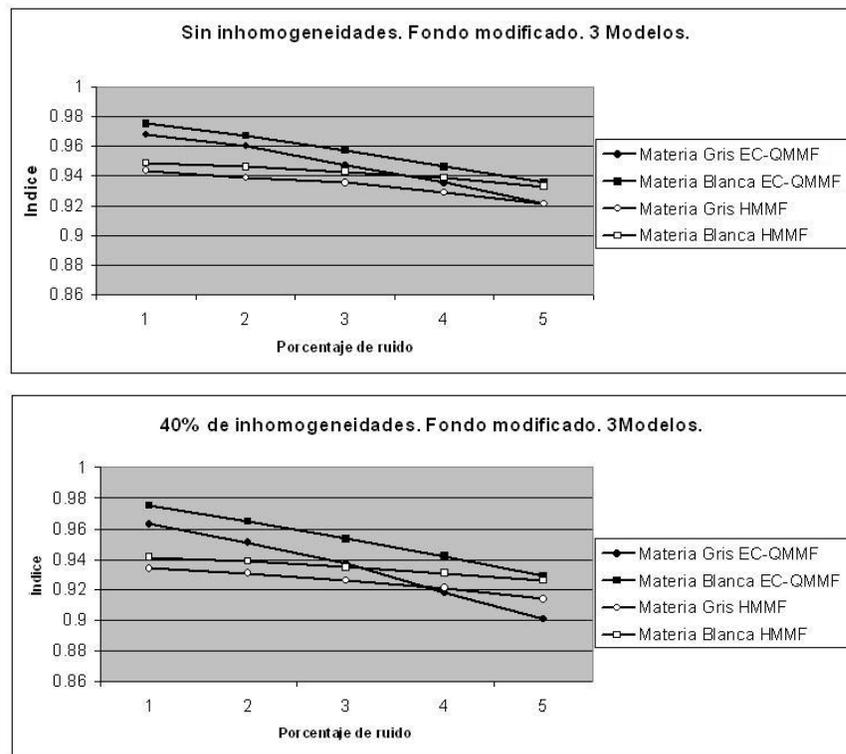


Figura 7.14: Resultados de la segmentación de imágenes de resonancia magnética del cerebro (simulaciones de Montreal). Segmentación usando 3 modelos sobre las imágenes con fondo modificado

7.5. Estimación del flujo óptico

En esta aplicación, estamos interesados en modelar el movimiento de los objetos en una secuencia de imágenes. Típicamente se asume que los objetos conservan su brillo (nivel de gris) a lo largo de su movimiento. Para utilizar EC-QMMF es necesario elegir un conjunto discreto de posibles direcciones $\{d_k\}$ que pueden seguir los objetos. Un modelo de observación razonable es

$$g_0(r) = g_1(r + d_{f(r)}) + \eta_r$$

donde $\eta_r \sim N(0, \sigma^2)$, g_0 y g_1 son dos imágenes consecutivas de la secuencia observada y $d_{f(r)}$ es el desplazamiento del sitio $r \in \mathcal{S}$. Es decir, el brillo de los objetos se conserva a lo largo de su movimiento salvo por pequeñas variaciones aleatorias. La log-verosimilitud (en un punto $r \in \mathcal{S}$) es entonces

$$\log P(g_1(r), g_0(r) \mid f_r = q_k) = -\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}(g_0(r) - g_1(r + d_k))^2$$

En general los componentes de d_k no son necesariamente enteros por lo que podría ocurrir que $r + d_k \notin \mathcal{S}$ y por tanto $g_1(r + d_k)$ puede no estar definido. Para resolver este problema utilizamos interpolación bicúbica de la imagen g_1 [8].

Los modelos $\{d_k\}$ se eligen tratando de cubrir lo mejor posible el rango de movimientos presentes en la imagen, pero dado que no contamos con información al respecto simplemente fijamos manualmente un desplazamiento máximo (en magnitud) R y un factor de discretización m . Elegiremos los desplazamientos dentro de la región dada por $[-R, R] \times [-R, R]$ discretizándola en $(2m + 1)^2$ modelos. Sea $\delta = \frac{R}{m}$, la discretización queda definida por los modelos

$$d_{i,j} = (i\delta, j\delta)^T \quad i, j = -m, -(m-1), \dots, 0, 1, \dots, m$$

donde el doble subíndice (i, j) fue introducido en lugar de k por claridad.

Notemos que una vez obtenida la expresión de la log-verosimilitud, el cálculo del campo de probabilidades p es idéntico que en los experimentos anteriores. En los siguientes experimentos se muestran los resultados usando modelos constantes de desplazamiento. Además omitimos el paso de estimación de parámetros, es decir los desplazamientos $\{d_k\}$ se mantienen fijos en todo el proceso. Usaremos la secuencia del Taxi de Hamburgo (Figura 7.15) para probar nuestro algoritmo.

El potencial a minimizar es el siguiente

$$U(p, d) = \sum_{r \in \mathcal{S}} \sum_{i=1}^K ((g_0(r) - g_1(r + d_i))^2 - \mu) p_i(r)^2 + \\ + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} \sum_{i=1}^K (p_i(r) - p_i(s))^2$$



Figura 7.15: Imágenes 0 y 19 de la secuencia del taxi de Hamburgo.

Una vez calculado el campo de probabilidades podemos elegir diferentes estimadores del flujo óptico. Para cada sitio r tenemos la distribución discreta $p(r)$, por lo que podemos obtener por ejemplo el valor esperado del desplazamiento en r respecto a la distribución $p(r)$, es decir, elegimos como desplazamiento en r el vector

$$v(r) = E_{p(r)}[d] = \sum_{i=1}^K d_i p_i(r)$$

llamaremos a este estimador el *desplazamiento medio* en el sitio r . Más en general podemos elegir los estimadores

$$v_n(r) = \sum_{i=1}^K d_i p_i(r)^n$$

y por supuesto el estimador MPM que usamos en segmentación:

$$v_{MPM}(r) = d_k, k = \operatorname{argmax}_i \{p_i(r)\}$$

En este caso no tenemos forma de estimar σ como lo hicimos en la segmentación por niveles de gris, entonces utilizaremos la variante en la que incorporamos este factor en los parámetros μ, λ . En la figura 7.16 se muestran los resultados obtenidos usando el método antes descrito usando las imágenes 0 y 1 de la secuencia del taxi de Hamburgo, es interesante notar que el movimiento del taxi está formado por una combinación de diferentes movimientos, esto se debe a que estamos modelando el complejo movimiento del taxi (un giro) mediante traslaciones constantes cuyos modelos, además, permanecen fijas a lo largo de todo el proceso. En la figura a) se muestra la velocidad del estimador MPM, en la figura b) la velocidad del flujo medio, en c) la velocidad del flujo v_n con $n = 2$ y en d) se muestran las regiones con

desplazamiento estimado distinto de cero (las regiones en movimiento).

Los parámetros fueron $\lambda = 0.02$, $\mu = 0.008$, el número de modelos utilizados fue 49 ($m = 3$) y el número de iteraciones 800 (solo se utilizó el criterio del número de iteraciones para detener el proceso).

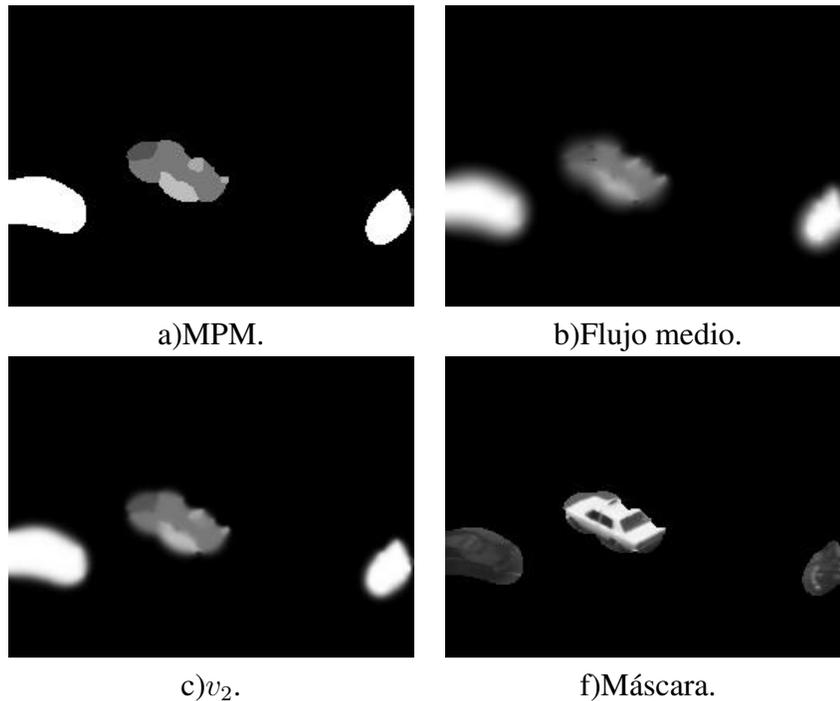


Figura 7.16: Estimación del flujo óptico usando el modelo EC-QMMF sobre las dos primeras imágenes de la secuencia del taxi de Hamburgo (ver texto).

Un problema común en los métodos de estimación de flujo óptico de este estilo que incluyen términos de regularización del estilo

$$\lambda \sum_{\langle r,s \rangle \in \mathcal{C}_2} \|p(r) - p(s)\|^2$$

es el que se observa en la figura 7.16: el flujo óptico estimado continúa alrededor de los objetos en movimiento.

Una idea inmediata para tratar de resolver este problema es hacer que los bordes de movimiento coincidan con los bordes de las imágenes. Para hacer esto primero necesitamos encontrar los bordes de las imágenes. Podemos utilizar el mismo modelo aproximado de campos ocultos para segmentar las imágenes (en realidad solamente necesitamos segmentar g_0), los bordes quedan determinados por la

segmentación. En la figura 7.17 se muestran los bordes obtenidos usando 5 modelos para la segmentación de la imagen g_0 , los parámetros usados en esta segmentación fueron $\lambda = 0.01$, $\mu = 0.015$.

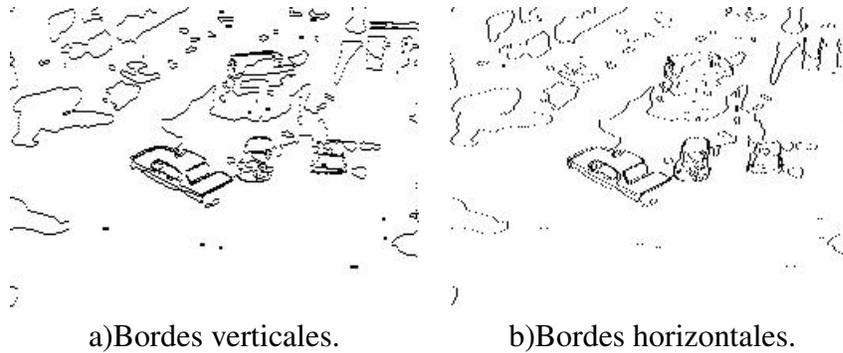


Figura 7.17: Bordes utilizados para la estimación del flujo óptico (ver texto).

El potencial modificado tiene la forma

$$U(p, d) = \sum_{r \in \mathcal{S}} \sum_{i=1}^K (\log v_i(r) - \mu) p_i(r)^2 + \\ + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} b(r, s) \|p(r) - p(s)\|^2$$

donde $b(r, s) = \delta(c_r - c_s)$ en la segmentación de g_0 , es decir, es igual a 1 si r y s pertenecen a la misma clase (de la segmentación de g_0) y es igual a 0 si no. Notemos que una vez hecha la segmentación de g_0 , las variables $b(r, s)$ son fijas.

En la figura 7.18 mostramos los resultados del método antes descrito. Al igual que en la figura 7.16 se muestra la velocidad del estimador MPM(a), flujo medio(b), v_n con $n = 2$ (c) y las regiones con desplazamiento no nulo(d).

Podemos observar que en estos resultados los bordes de movimiento están muy bien definidos, aunque también se aprecia un problema relacionado con la sobre-segmentación de g_0 , hay algunas regiones sin movimiento que son detectadas como regiones con movimiento no nulo, este problema se corregía antes con la regularización completa. Todo esto sugiere que lo que requerimos es realizar las segmentaciones (de nivel de gris y de movimiento) simultáneamente. Una opción es simplemente elegir $b(r, s) \in [0, 1]$ en lugar de $b(r, s) \in \{0, 1\}$ para lo cual también podemos usar los resultados de EC-QMMF de la segmentación de g_0 por nivel de gris (una medida relacionada con la entropía de p podría resultar).

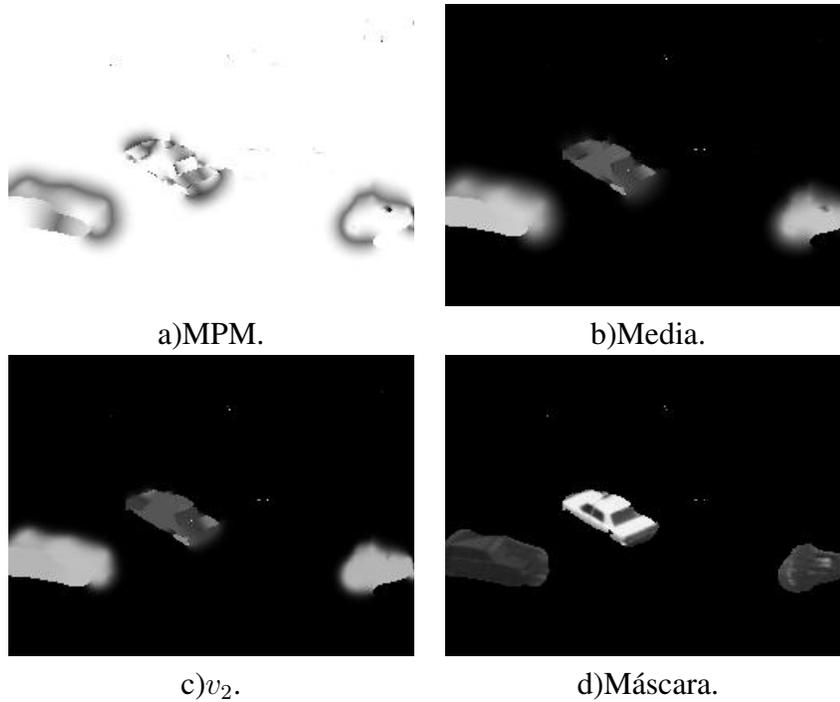


Figura 7.18: Estimación del flujo óptico introduciendo información de los bordes(ver texto).

Una variante interesante es la de ajustar un flujo afín a cada objeto en movimiento, la función de energía a minimizar para la estimación del flujo afín es

$$U(A, b) = \sum_{r \in \mathcal{W}} (g_0(r) - g_1(Ar + b))^2$$

aquí utilizamos la notación $r = (x, y)^T$ para representar la transformación afín en forma matricial. \mathcal{W} es la región de la imagen sobre la que deseamos calcular el flujo afín. La minimización de este potencial se realiza sobre los componentes de la matriz A de 2×2 y el vector b . Al derivar la energía e igualar a cero obtenemos un sistema de 6 ecuaciones lineales con 6 incógnitas que puede ser resuelto fácilmente usando cualquier método tradicional como descomposición LU o gradiente conjugado [8][30]. En esta aplicación en particular utilizamos gradiente conjugado.

En la figura 7.19 se muestra un experimento de esta variante, el principal problema es que se requiere una segmentación razonablemente buena de los objetos en movimiento.

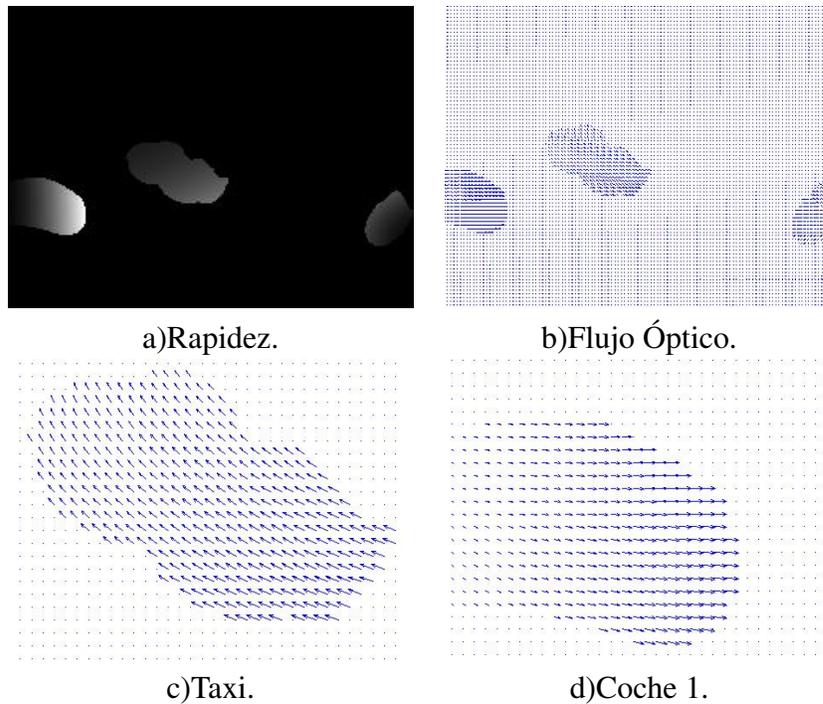


Figura 7.19: Flujo óptico estimado mediante el modelo afín para cada objeto en movimiento.

7.5.1. Discusión

De acuerdo a los resultados mostrados en esta sección podemos concluir que aunque EC-QMMF por sí solo puede dar aproximaciones relativamente buenas del flujo óptico, es necesario mejorar muchos aspectos del modelo para obtener resultados sobresalientes. Quedan abiertos muchos problemas por resolver entre los que puedo citar los siguientes:

- Diseñar un modelo con el que podamos lograr que los bordes de movimiento coincidan con los bordes por niveles de gris.
- Elaborar un método de selección de modelos con el que se puedan eliminar modelos inutilizados para acelerar el proceso.
- Incorporar el ajuste de los modelos(en este trabajo los modelos permanecieron fijos).
- Encontrar una buena manera de elegir los modelos iniciales.

7.6. Restauración

En esta sección mostramos la aplicación de EC-QMMF al problema de restauración de imágenes. Este problema ya fue planteado y estudiado a grandes rasgos en el capítulo 3, por lo que pasaremos directamente al método basado en EC-QMMF y los resultados experimentales.

Aprovechando el hecho de que nuestro método de segmentación nos da una estimación de los modelos paramétricos que generaron la imagen original, f , donde

$$f_r = \phi_{c_r}(r)$$

y que además contamos con el campo de medida

$$p_k(r) = P(c_r = q_k)$$

lo cual equivale a

$$p_k(r) = P(f_r = \phi_{c_k}(r))$$

podemos entonces calcular diferentes estimadores para f_r usando la distribución discreta $p(r)$. Vamos a ver las propiedades de tres diferentes estimadores:

- Moda

$$\hat{f}_r = \phi_k(r)$$

donde

$$k = \operatorname{argmax}_i \{p_i(r)\}$$

- Media

$$\bar{f}_r = E_{p(r)}[f_r]$$

- Mediana

$$\tilde{f}_r = x$$

donde

$$P(f_r \leq x) = \frac{1}{2}$$

Para una distribución discreta no podemos, en general, calcular la mediana, pero usaremos una aproximación lineal. Si $p_1(r) \geq 0.5$ entonces la mediana es justamente $\phi_1(r)$. De otro modo, sean $p =$

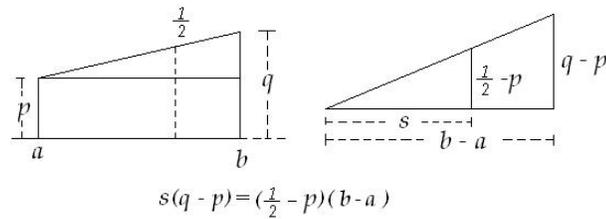


Figura 7.20: Aproximación lineal de la mediana en distribuciones discretas (ver texto).

$p_j(r)$, $q = p_{j+1}(r)$ los elementos de la distribución discreta $p(r)$ tales que $p < \frac{1}{2} \leq q$, entonces definiendo $a = \phi_j(r)$, $b = \phi_{j+1}(r)$ y usando la regla de Tales (ver figura 7.20)

obtenemos la aproximación lineal

$$\tilde{f}_r = a + s = a + \left(\frac{1}{2} - p\right) \frac{(b-a)}{(q-p)}$$

Para aplicar EC-QMMF para el problema de restauración, vamos a calcular el campo de medida p considerando una cantidad suficientemente grande de **modelos constantes**, en los siguientes experimentos utilizamos 15 modelos constantes, tomamos la conocida imagen de Lenna (normalizada en el rango $[0, 1]$) contaminada con ruido Gaussiano con $\sigma = 0.05$, la *restauración* es la imagen f formada por los estimadores de \tilde{f}_r correspondientes, ya sea la moda, la media o la mediana.

En la figura 7.22 se muestra el resultado del filtrado al utilizar diferente cantidad de modelos con EC-QMMF. Es interesante ver cómo a pesar de que la mediana presenta mejores resultado “visualmente” (las imágenes se ven mejor) el error cuadrático medio es superior al de la media.



Figura 7.21: Diferentes estimadores de f usando las distribuciones discretas $p(r)$ en cada sitio r . a)Imagen contaminada con ruido Gaussiano con varianza 0.0025. b)Estimador MPM, el error cuadrático medio (MSE por Mean Squared Error) es 0.0011. c)Media, el MSE es 0.0011. d)Mediana, el MSE es 0.0015.

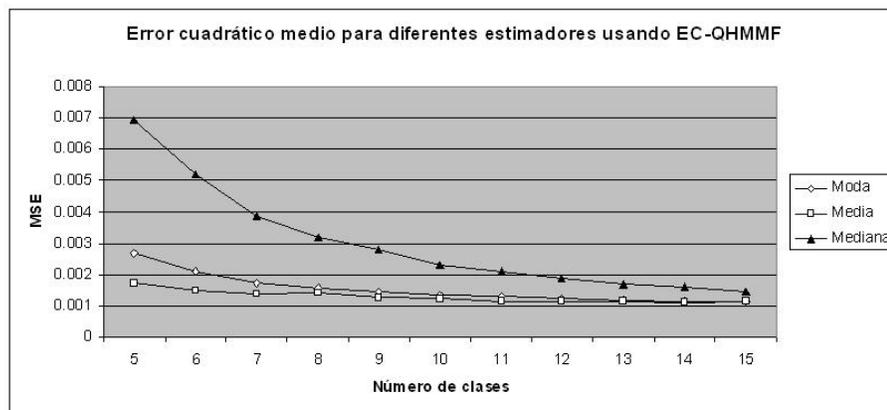


Figura 7.22: Error cuadrático medio para diferentes estimadores usando EC-QMMF(ver texto).

7.7. Segmentación de imágenes provenientes de colposcopías

En la siguiente aplicación utilizamos datos provenientes de colposcopías, que son secuencias de imágenes del cérvix que se toman para captar las variaciones en el color del tejido como reacción al ácido acético. El tejido dañado por papiloma reacciona al ácido acético haciéndose más claro que el tejido sano para posteriormente volver a su tonalidad base. La curva que describe el cambio en la coloración del tejido respecto al color base sirve como criterio para discriminar entre tejido sano y tejido dañado.

En la primera sección se describe un primer enfoque para la solución de este problema, el cual fue desarrollado por investigadores de la Universidad Veracruzana. En este primer enfoque, luego del pre-proceso de las imágenes que consiste básicamente en filtrado y alineamiento de las imágenes del video, cada sitio de la región de interés se clasifica de manera independiente por un clasificador Bayesiano ingenuo entrenado con un conjunto de datos que fue etiquetado por un experto. Las categorías son: tejido normal, metaplasia inmadura (zonas de transición entre tejido normal y dañado), lesión de bajo grado y lesión de alto grado. Los rasgos que se utilizan para la clasificación son los componentes del vector que resulta de la proyección de cada serie de tiempo sobre las primeras componentes principales.

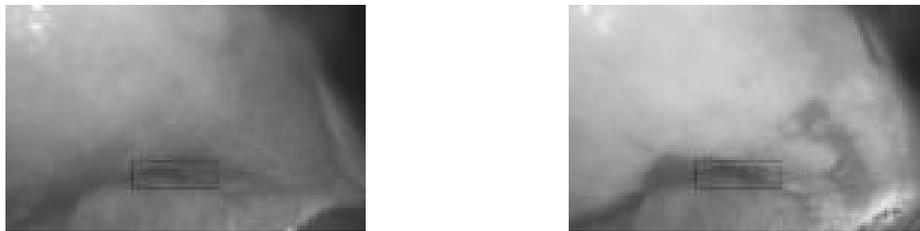
Un segundo enfoque, presentado en la segunda sección fue desarrollado durante una estancia en la Universidad Veracruzana. El objetivo era únicamente el de incorporar EC-QMMF como método de segmentación reemplazando al clasificador Bayesiano ingenuo para considerar la interacción espacial. El pre-proceso de las imágenes no varía en este segundo enfoque. La información dada por el experto no se utiliza en esta primera modificación. En lugar de reducir la dimensionalidad mediante componentes principales se propone un modelo que considera las series de tiempo completas.

En la tercera sección se estudian posibles modificaciones al proceso de clasificación para imágenes de colposcopia y posibles extensiones de EC-QMMF para incluir modelos de observación implícitos mediante conjuntos de entrenamiento, esta idea surge de las aplicaciones en las que el modelo de observación es difícil de definir y un enfoque de aprendizaje automático (aprendizaje máquina o estadístico) es más práctico.

7.7.1. Trabajo previo

Todo el procedimiento descrito en esta sección fue desarrollado por los doctores Héctor Gabriel Acosta, Sebastián Camacho, Nicandro Cruz entre otros investigadores de la Universidad Veracruzana.

En la figura 7.23 se muestran ejemplos del tipo de imágenes en las secuencias de colposcopías. Aparentemente, la respuesta al ácido acético es lo suficientemente contrastante para que algún procesamiento sencillo pueda discernir entre dos tipos de tejidos. Sin embargo en las imágenes se encuentran presentes algunos artefactos que ocasionan que en algunas regiones se tengan altos niveles de ruido. Además, el movimiento de la cámara y la paciente es inevitable durante la adquisición del video, así que es necesario aplicar un preprocesamiento de registro para alinear las imágenes. El procedimiento se divide en cuatro procesos principales: registro, extracción de rasgos, extracción de conocimiento (del experto), Aprendizaje Bayesiano y segmentación [18].



(a) Antes de la aplicación del ácido acético. (b) 32 segundos después de la aplicación.

Figura 7.23: Imágenes de una colposcopia. El rectángulo negro se utilizó como referencia durante las pruebas de preprocesamiento, pero no se utilizó en el proceso final.

El registro se realiza en dos partes, primero se aplica un registro global suponiendo que la deformación es una traslación y aplicando un método basado en correlación de fase [12] para después refinar el registro usando correlación cruzada normalizada [25]. Una vez registradas las imágenes es posible extraer las series de tiempo asociadas a cada sitio (figura 7.24), a cada una de estas series de tiempo se les llama Función de Respuesta al Ácido Acético (AWRF por “Aceto-White Response Function”).

Posteriormente se le solicitó a un especialista que señalara manualmente algunas regiones del tejido que presentara diferentes grados de lesión. El especialista señaló regiones correspondientes a cuatro categorías: tejido normal (figura 7.25(a)), *metaplasia inmadura* o regiones de transición entre tejido normal y una lesión (figura 7.25(b)), lesión de bajo grado (figura 7.25(c)) y lesión de alto grado (figura 7.25(d)). Estos datos son útiles para estudiar las diferencias que definen el grado de lesión.

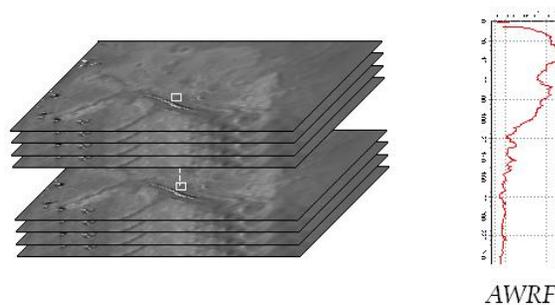


Figura 7.24: Una vez registradas las imágenes, es posible definir para cada sitio, una serie de tiempo(ver texto).

El conjunto de datos consta de tres secuencias de video tomadas de pacientes diferentes, una de ellas fue utilizada por el especialista para realizar la segmentación manual (el especialista no segmentó la región completa, sino que señaló pequeñas subregiones que consideró características del grado de lesión en cuestión) (ver figura 7.25).

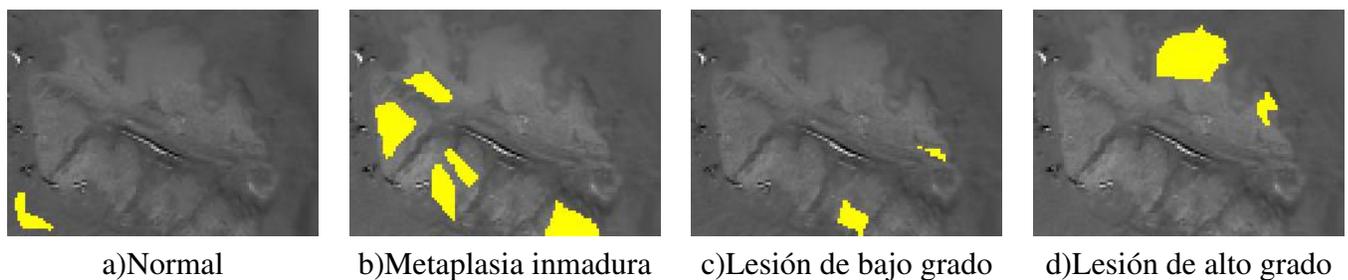


Figura 7.25: Segmentación manual realizada por el especialista(ver texto).

En la figura 7.26 se muestra la serie de tiempo asociada a un sitio de cada región segmentada por el especialista. A primera vista es muy difícil determinar las características que definen los cuatro tipos de tejido, sin embargo parece ser relativamente sencillo discernir entre tejido normal y el resto, el comportamiento corresponde al descrito antes, en las regiones con lesión de algún grado el brillo aumenta en la primera parte de la serie de tiempo para después regresar a la tonalidad base, mientras que en las regiones correspondientes a tejido normal el brillo se mantiene en la tonalidad base. Además podemos ver que en general las series de tiempo no son para nada suaves sino que presentan una cantidad relativamente alta de ruido.

En resumen, contamos con un conjunto de datos $X = \{x_i\} \subset \mathbb{R}^n$, donde n es el número de imágenes en la secuencia de video. Cada x_i es una serie de tiempo asociada a un sitio en particular. Contamos

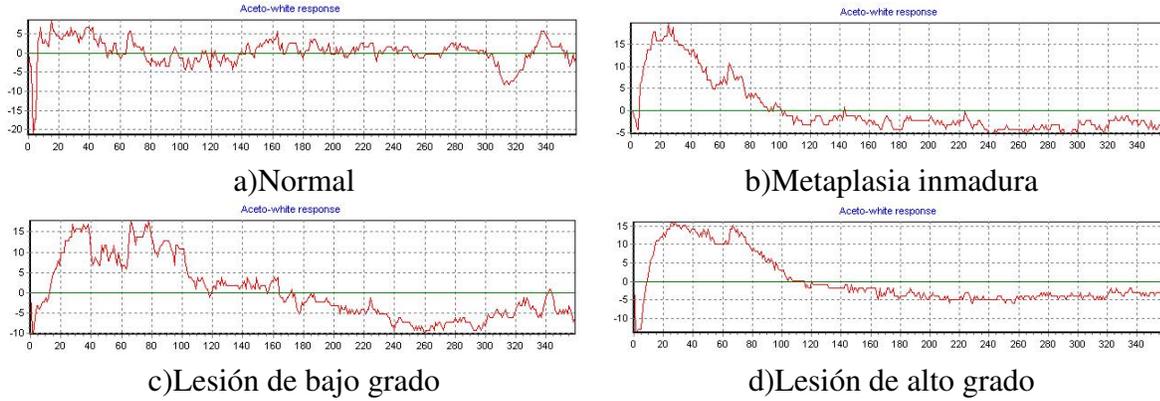


Figura 7.26: Ejemplos de series de tiempo asociadas a las cuatro diferentes categorías.

además con las etiquetas sociadas a cada sitio, para un subconjunto $Z = \{z_i\} \subset X$.

Con el objetivo de reducir la dimensión de los datos y tratar de capturar las características globales de los datos, se realizó un análisis de *componentes principales* [19](PCA por *Principal Component Analysis*).

El objetivo del método de componentes principales, en pocas palabras, es encontrar una transformación lineal $P : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $p \ll n$, tal que los datos transformados bajo P capturen la información de los datos originales lo mejor posible. Más formalmente, estamos interesados en encontrar la matriz M de $p \times n$ tal que

$$\sum_{i=1}^N \|x_i - M^T M x_i\|^2$$

sea mínimo, donde N es la cardinalidad del conjunto de datos. No es difícil probar que la matriz que minimiza la expresión anterior es aquella cuyas p filas son los vectores propios de la matriz de covarianza estimada a partir de X , es decir Σ donde

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T$$

donde $\mu = \frac{1}{N} \sum_{i=1}^N x_i$.

En el caso particular de esta aplicación, contamos con datos provenientes de tres pacientes (tres videos), cada secuencia consta de 360 imágenes ($n=360$), de 88×121 píxeles ($N = 3 \times 88 \times 121 = 31,944$). Los primeros cinco componentes principales ($p = 5$) describen más del 95% de la varianza.

Finalmente, usando los datos transformados $\hat{X} = \{\hat{x}_i = M^T M x_i\}$, se utiliza un clasificador *Bayesiano ingenuo* (naive Bayes) para la clasificación.

El clasificador Bayesiano ingenuo se basa en la suposición de que dada la clasificación, los rasgos son independientes entre si, esto es, sea $x \in \mathbb{R}^p$ una observación en nuestro espacio reducido usando componentes principales, llamaré a cada componente de x un “rasgo”, sea c la clase a la que pertenece la observación x , entonces

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

la suposición de independencia es

$$P(x | c) = \prod_{i=1}^p P(x_i | c)$$

A partir de esto hay diversas variantes tanto para calcular $P(x_i | c)$ (la verosimilitud) como para establecer $P(c)$ (información a priori). En general, para calcular la verosimilitud se establece algún modelo paramétrico para ésta y los parámetros se ajustan usando los datos. Se invita al lector interesado a revisar los detalles en las referencias [29] [18].

Entre las principales ventajas de este clasificador están la eficiencia de entrenamiento, la sencillez del modelo y la posibilidad de incorporar la información dada por el especialista.

Las principales desventajas son que la clasificación no considera el contexto espacial (cada sitio se clasifica de manera independiente) por lo que la segmentación final presenta una granularidad relativamente alta. En la figura 7.27 se muestra la segmentación obtenida usando el clasificador Bayesiano ingenuo en una de las secuencias, a la izquierda se muestra una de las imágenes, preprocesadas, de dicha secuencia. La siguiente desventaja, es que el modelo asume la independencia de las variables de entrada dada la clasificación, lo cual es, en general, poco realista. Sin embargo es importante señalar que en la literatura se reportan excelentes resultados al utilizar este clasificador en los que la suposición de independencia no es algo grave.

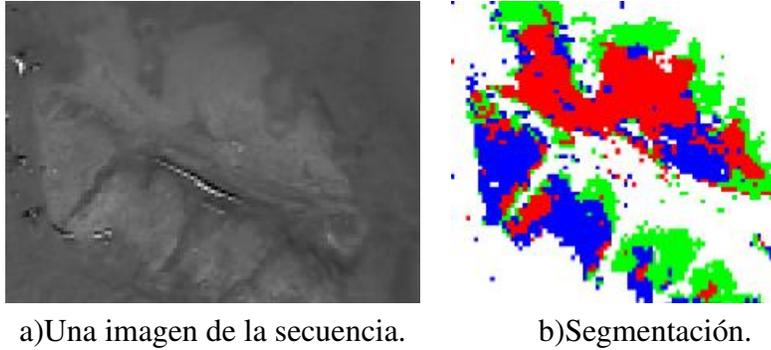


Figura 7.27: Segmentación obtenida usando el clasificador Bayesiano ingenuo. Rojo: lesión de alto grado. Verde:Lesión de bajo grado. Azul:metaplasia inmadura.

7.7.2. Aplicación de EC-QMMF para series de tiempo

En secciones anteriores, al describir el método de segmentación EC-QMMF, mostramos que básicamente el único *ingrediente* necesario para aplicar dicho método a un problema de segmentación específico es contar con una manera de calcular $v_k(r)$, la verosimilitud. En esta aplicación se nos presenta un problema en el que la verosimilitud está dada de manera indirecta mediante una *conjunto de entrenamiento* a partir del cual necesitamos inferir la forma de la verosimilitud.

Tomando los datos en el espacio reducido usando componentes principales $\hat{X} = \{x_i\}$ podemos formar $\frac{p(p-1)}{2}$ gráficas componente a componente donde p es el número de componentes de cada vector en \hat{X} . En cada cuadro de la figura 7.28 se muestra una de estas gráficas. Estamos interesados en encontrar cúmulos bien definidos de cada clase, de esta forma un procedimiento de aprendizaje estadístico no tendrá problemas para *aprender* a clasificar los puntos. El resultado no es alentador, vemos que los datos están extremadamente mezclados en cualquier dirección (lo cual, sin embargo, no prueba que no hay una separación en el espacio reducido). Este resultado sugiere que es preferible trabajar con las series de tiempo completas en lugar de usar solamente los coeficientes definidos por la proyección sobre los componentes principales.

La desventaja es que en su versión “pura”, EC-QMMF no contempla la incorporación de la verosimilitud de manera implícita sino a través de un modelo de observación, así que la forma de proceder será definir un modelo de observación en lugar de usar aprendizaje estadístico para posteriormente asociar las clases de la segmentación a las categorías de interés.

Supongamos que cada serie de tiempo se puede representar, aproximadamente, como combinación

lineal de un conjunto pequeño de funciones base

$$g_r(t) \approx \sum_{i=1}^K \theta_i f_i(t) \quad \forall t = 0, 1, \dots, T$$

donde estas funciones base son fijas. Usaremos precisamente las componentes principales ya que las series de tiempo pueden representarse fielmente usando solamente estas funciones. El modelo de observación es el siguiente

$$g_r(t) = \sum_{i=1}^K \theta_i f_i(t) + \eta_r(t)$$

donde $\eta_r(t) \perp \eta_s(u) \forall t \neq u, r \neq s$. En la figura 7.29 se muestran dos series de tiempo típicas y sus ajustes usando componentes principales para mostrar la factibilidad de este modelo

Bajo este modelo, podemos expresar la verosimilitud como

$$v_k(r) = \prod_{t=0}^T P(g_r(t) | c_r = k) = \prod_{t=0}^T \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{1}{2\sigma^2} (g_r(t) - \sum_{i=1}^K \theta_i^{(k)} f_i(t))^2}$$

y la log-verosimilitud como

$$\log v_k(r) = -(T+1) \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \sum_{t=0}^T \left(g_r(t) - \sum_{i=1}^K \theta_i^{(k)} f_i(t) \right)^2$$

Como vimos en un principio, dado que finalmente vamos a optimizar respecto a p, θ , podemos incorporar el término $-(T+1) \log(\sqrt{2\pi}\sigma)$ en μ y el factor $\frac{1}{2\sigma^2}$ en λ para obtener el potencial

$$U(p, \theta) = \sum_{r \in \mathcal{S}} \sum_{k=1}^K \left(\sum_{t=0}^T \left(g_r(t) - \sum_{i=1}^K \theta_i^{(k)} f_i(t) \right)^2 - \mu \right) p_k(r)^2 + \lambda \sum_{\langle r, s \rangle \in \mathcal{C}_2} \sum_{i=1}^K (p_i(r) - p_i(s))^2$$

Para optimizar respecto a $\theta_j^{(k)}$:

$$\frac{\partial U}{\partial \theta_j^{(k)}}(p, \theta) = 2 \sum_{r \in \mathcal{S}} p_k(r)^2 \sum_{t=0}^T \left(\sum_{i=1}^K \theta_i^{(k)} f_i(t) - g_r(t) \right) f_j(t) = 0$$

Definamos

$$s_{i,j} = \sum_{t=0}^T f_i(t) f_j(t)$$

y

$$s_i(r) = \sum_{t=0}^T f_i(t) g_r(t)$$

entonces

$$\sum_{r \in \mathcal{S}} p_k(r)^2 \sum_{i=1}^K \theta_i^{(k)} s_{i,j} = \sum_{r \in \mathcal{S}} p_k(r)^2 s_j(r) \Rightarrow \sum_{i=1}^K \theta_i^{(k)} s_{i,j} = \frac{\sum_{r \in \mathcal{S}} p_k(r)^2 s_j(r)}{\sum_{r \in \mathcal{S}} p_k(r)^2}$$

lo cual define, para cada modelo $k = 1, 2, \dots, K$ un sistema de ecuaciones lineales:

$$\begin{bmatrix} s_{1,1} & s_{2,1} & \dots & s_{K,1} \\ s_{1,2} & s_{2,2} & \dots & s_{K,2} \\ \dots & & & \\ s_{1,K} & s_{2,K} & \dots & s_{K,K} \end{bmatrix} \begin{bmatrix} \theta_1^k \\ \theta_2^k \\ \dots \\ \theta_K^k \end{bmatrix} = \frac{1}{\sum_{r \in \mathcal{S}} p_k(r)^2} \sum_{r \in \mathcal{S}} p_k(r)^2 \begin{bmatrix} s_1(r) \\ s_2(r) \\ \dots \\ s_K(r) \end{bmatrix}$$

Realizamos la segmentación usando cuatro modelos, los resultados se muestran en la figura 7.30. Los modelos se inicializaron usando el rango dinámico de las imágenes, simplemente fijamos

$$s_k(t) = m(t) + (k - 1) \frac{M(t) - m(t)}{p - 1} \quad k = 1, 2, \dots, p$$

donde $m(t)$ es el valor mínimo en la imagen t en la secuencia y $M(t)$ el valor máximo.

A pesar de que la segmentación es un poco menos granular, este segmentador no fue capaz de diferenciar los sitios de metaplasia inmadura con los correspondientes a lesión de bajo grado, lo cual sí fue posible al utilizar el segmentador Bayesiano ingenuo.

7.7.3. Posibles modificaciones y extensiones

Un aspecto particularmente interesante es la cuestión de cómo incorporar la información de manera implícita mediante ejemplos. A pesar de que al usar componentes principales no se aprecia una separación clara, otros métodos podrían ser útiles. Un ejemplo es el método propuesto por Mu Zhu y Trevor Hastie [36]. Este interesante método es una generalización del método de Fisher (LDA por *Linear Discriminant Analysis* [13]) en el cual se asume que las distribuciones de los datos son normales. La primera dirección discriminante de Fisher está dada por

$$\operatorname{argmax}_{\alpha \in \mathbb{R}^n} \frac{\alpha^T B \alpha}{\alpha^T W \alpha}$$

donde B es la matriz de covarianza *entre clases* (la covarianza estimada usando todos los datos) y W es la matriz de covarianza *dentro de las clases* (típicamente un promedio de las covarianzas estimadas para

cada clase por separado). Dadas las primeras $(m - 1)$ direcciones discriminantes, la m -ésima dirección es simplemente

$$\operatorname{argmax}_{\alpha \in \mathbb{R}^n} \frac{\alpha^T B \alpha}{\alpha^T W \alpha}$$

sujeta a

$$\alpha^T W \alpha = 0 \forall j < m.$$

usualmente el uso de direcciones discriminantes mejoran el desempeño de clasificadores sobre los conjuntos de prueba, sin embargo el uso de LDA no siempre asegura la obtención de las mejores direcciones.

La aportación de Zhu y Hastie parte de analizar el criterio de Fisher desde el punto de vista de la verosimilitud. Primero, supongamos que dado $y = k$ (la clase a la que pertenece x es k), la densidad de x está dada por p_k , ahora consideremos la siguiente prueba de hipótesis

$$\begin{aligned} H_0 : & p_k = p \forall k = 1, 2, \dots, K \\ H_A : & p_k \neq p \text{ para algún } k = 1, 2, \dots, K \end{aligned}$$

Una manera natural de contrastar la hipótesis nula es usando la razón de verosimilitudes

$$LR(\alpha) = \log \frac{\max_{p_k} \prod_{k=1}^K \prod_{x_j \in C_k} p_k^{(\alpha)}(\alpha^T x_j)}{\max_{p_k=p} \prod_{k=1}^K \prod_{x_j \in C_k} p^{(\alpha)}(\alpha^T x_j)}$$

Para ver por qué esta es una buena medida de la efectividad de α como dirección discriminante, nótese que al proyectar los datos sobre una *mala* dirección discriminante, los datos estarán muy mezclados, es decir, no habrá una separación clara entre ellos, por lo que la hipótesis nula no podrá ser rechazada. Por otro lado, al tomar una *buen*a dirección discriminante, al menos una de las clases estará bastante separada de las demás por lo que la hipótesis nula será rechazada. Para implementar esto, utilicé estimaciones de las densidades (univariadas!) de p y p_k basadas en kernels (o *ventanas de Parzen* [13]), finalmente el problema se traduce a un problema de optimización con restricciones, ya que para encontrar más de una dirección discriminante es necesario restringir el espacio de búsqueda, en el trabajo de Zhu y Hastie se sugieren dos maneras de encontrar direcciones discriminantes adicionales. Se invita al lector interesado a seguir los detalles de este método en las referencias [36], donde además se propone un método de optimización muy eficiente para atacar este problema, aunque el problema puede ser resuelto de manera eficiente usando cualquier método de optimización con restricciones [30]. Lo importante es señalar que usando este método es posible encontrar direcciones discriminantes en las que los cúmulos están

mejor definidos y separados. En la figura 7.31 se muestran las gráficas análogas a las de la figura 7.28 pero esta vez usando el método de Zhu y Hastie (FENPDA por *Feature Extraction for Non-Parametric Discriminant Analysis*).

Es interesante ver que en este caso las observaciones correspondientes al tejido normal se diferencian bastante bien del resto. Además los cúmulos correspondientes a metaplasia inmadura y lesión de alto grado están muy bien definidos, lo interesante es que justamente estas dos clases que se encuentran tan separadas corresponden a extremos en cuanto al nivel de lesión y al mismo tiempo se observa que lo que el especialista segmentó como lesión de bajo grado se confunde con lo que segmentó como región de transición (metaplasia inmadura). En resumen, usando estas direcciones discriminantes, un clasificador basado en aprendizaje estadístico no debería tener problemas en clasificar tres tipos de tejido: normal, lesión de bajo grado y lesión de alto grado. Uno de estos clasificadores combinado con EC-QMMF que incorpore el contexto espacial, en mi opinión promete buenos resultados.

7.7.4. Discusión

En esta sección se plantearon nuevos problemas e ideas que abren nuevas posibilidades en cuanto a la aplicación de EC-QMMF a diferentes problemas de segmentación. Se presentó un marco de aplicación para datos estructurados en altas dimensiones como son las series de tiempo y se dedujo el método general de ajuste y su solución.

La validación de los resultados obtenidos no puede llevarse a cabo de manera contundente debido a que se cuenta con muy pocos datos. No es suficiente simplemente dividir el conjunto de datos etiquetados manualmente en un conjunto de entrenamiento y uno de prueba porque provienen de la misma secuencia de video. Una validación contundente debe incluir conjuntos de entrenamiento y prueba provenientes de diferentes pacientes. Considero, sin embargo, que esta es una aplicación muy interesante que vale la pena estudiar más a fondo.

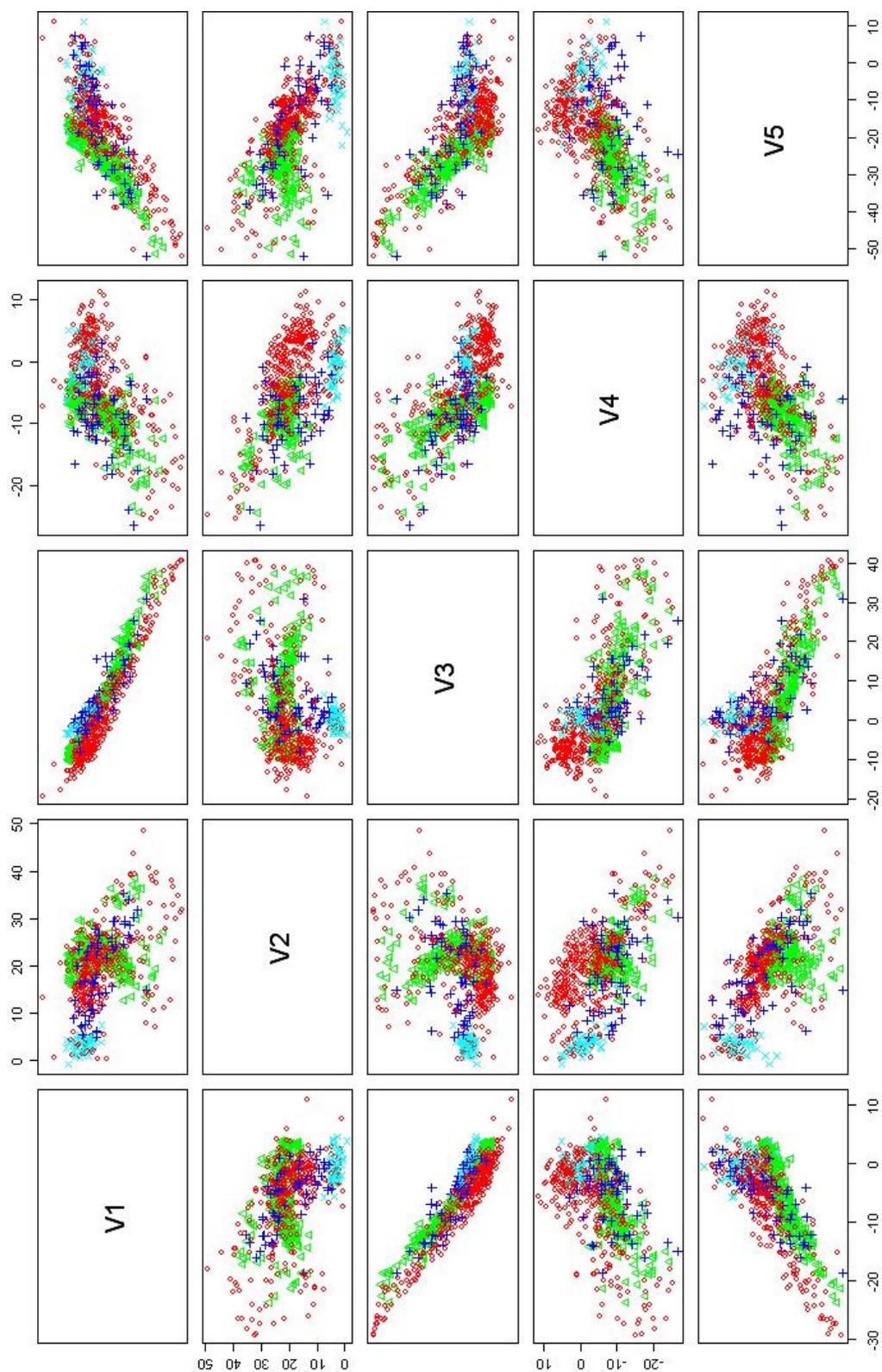


Figura 7.28: Coeficiente asociado a cada componente principal contra el resto (cada punto corresponde a un pixel de la clasificación manual).Circulos(rojo)=metaplasia inmadura. Triángulos(verde)=lesión de alto grado. +’s(azul)=lesión de bajo grado. x’s(cyan)=tejido normal.

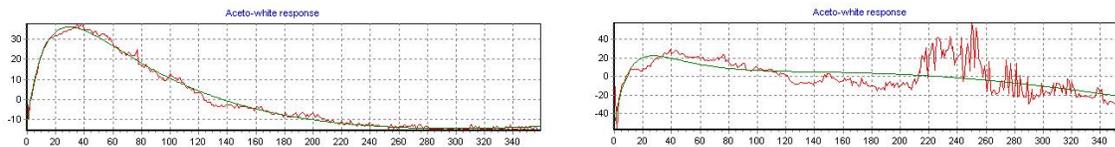


Figura 7.29: Ajuste obtenido usando las primeras 5 componentes principales. El ajuste es muy aceptable y muy robusto al ruido.

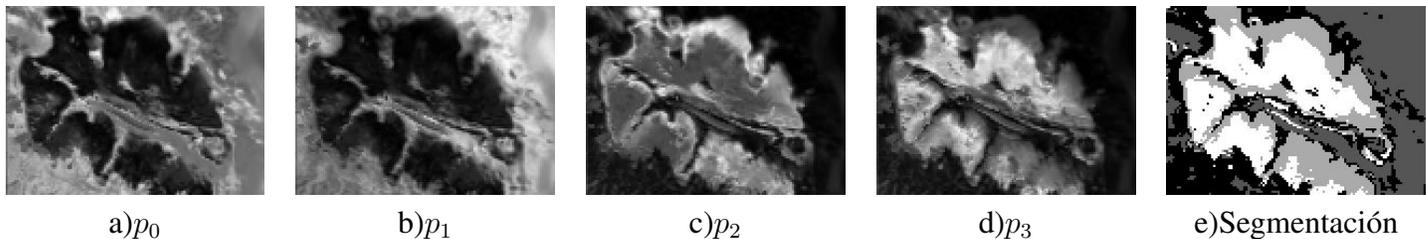


Figura 7.30: Resultados obtenidos en las imágenes de colposcopia. a)-d) la probabilidad de pertenencia de cada pixel a las clases 0, 1, 2, 3 y 4 respectivamente. Regiones claras representan alta probabilidad de pertenencia. e) La segmentación final usando el estimador MPM. Podemos asociar por ejemplo, las clases 0 y 1 a tejido normal y las clases 2 y 3 a tejido dañado.

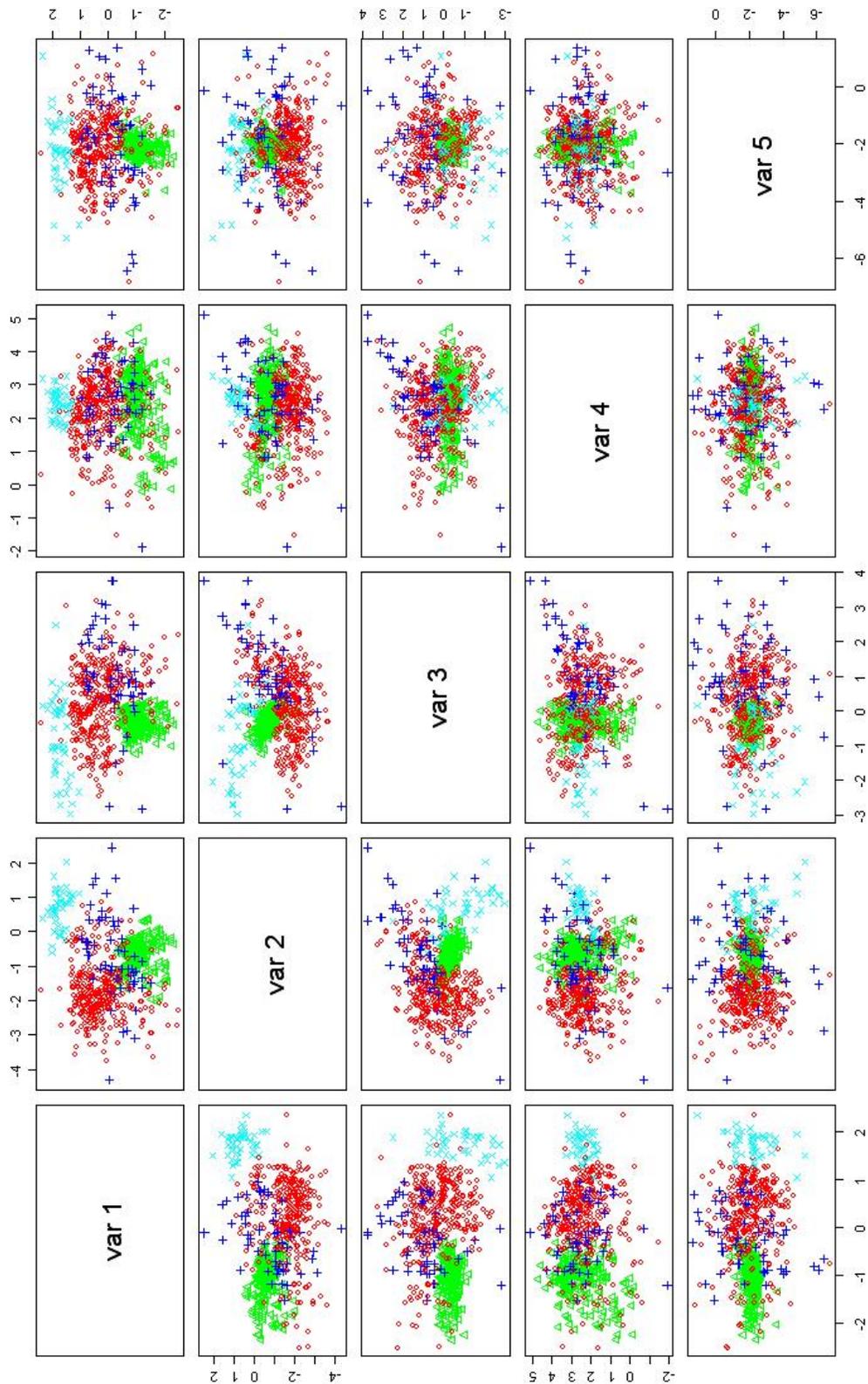


Figura 7.31: Coeficiente asociado a cada dirección encontrada usando el método FENPDA(ver texto) contra el resto (cada punto corresponde a un pixel de la clasificación manual).Círculos(rojo)=metaplasia inmadura. Triángulos(verde)=lesión de alto grado. +’s(azul)=lesión de bajo grado. x’s(cyan)=tejido normal.

Capítulo 8

Resumen y conclusiones

En este documento presentamos una aproximación cuadrática del potencial que surge del modelo de campos ocultos (HMMF). Entre las ventajas que proporciona el utilizar esta aproximación se encuentran:

- Es posible utilizar métodos de optimización muy eficientes y sencillos.
- Manejo directo de las restricciones lineales a través de un esquema iterativo de Gauss-Seidel.
- Control explícito de la entropía del campo de probabilidades estimado.
- El ajuste de los parámetros es muy sencillo, solamente se requiere ajustar dos parámetros cuya interpretación es clara.
- Uno de los parámetros está relacionado con la varianza del ruido presente en las imágenes por lo que es posible estimarlo a partir de los datos, con lo cual solo es necesario ajustar un parámetro. Más aún si el método se utiliza en imágenes de características similares, es posible obtener un método libre de parámetros, esto lo ilustramos con los resultados de la segmentación de cerebros.
- Se hereda toda la gama de aplicaciones de HMMF con la ventaja de una implementación más limpia y sencilla.

Finalmente, ilustramos la versatilidad del método aplicándolo a diferentes problemas de procesamiento de imágenes:

- Segmentación de imágenes de resonancia magnética usando modelos constantes (para el caso de imágenes sintéticas sin inhomogeneidades espaciales) y modelos de splines (imágenes sintéticas

con inhomogeneidades espaciales e imágenes reales de la base de datos de Harvard en las que también se aprecian inhomogeneidades espaciales). Los resultados experimentales superan por mucho a los métodos más actuales de segmentación.

- Estimación de Flujo Óptico usando modelos constantes de desplazamiento y fijos (sin estimar los modelos). Mostramos diferentes estimadores del flujo óptico a partir del campo de probabilidades p estimado.
- Segmentación a partir de características definidas por series de tiempo. Derivamos el procedimiento general para series de tiempo y mostramos la aplicación particular sobre imágenes de colposcopia.

Bibliografía

- [1] Edgar Arce, *Modelos de campos ocultos markovianos con aplicación a la estimación de la disparidad estereoscópica*, Ph.D. thesis, Centro de Investigación en Matemáticas, Departamento de Ciencias de la Computación, 2004.
- [2] Adrian Barbu and Song-Chun Zhu, *Incorporating visual knowledge representation in stereo reconstruction*, International Conference on Computer Vision (2005).
- [3] J. Besag, *On the statistical analysis of dirty pictures*, Journal of the Royal Statistical Society (1986), 259–302.
- [4] Michael J. Black and P. Anand, *A framework for the robust estimation of optical flow*, Proc. Fourth International Conference on Computer Vision (1993).
- [5] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE-PAMI (2001).
- [6] ———, *Energy minimization with discontinuities*, International Journal of Computer Vision (2003).
- [7] Yuri Boykov and Vladimir Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 26, no. 9 (2004), 1124–1137.
- [8] R.L. Burden and J.D. Faires, *Análisis numérico*, 1985.
- [9] H.E. Cline, C.L. Doumulin, H.R. Hart, W.E. Lorensen, and S.Ludke., *3-d reconstruction of the brain from magnetic resonance images using a connectivity algorithm.*, Magnetic Resonance Imaging (1987), 5:345–352.

- [10] C.A. Cocosco, V. Kollokian, R.K. Kwan, and A.C. Evans, *Online interface to a 3d mri simulated brain database.*, Neuroimage, 5(4):part 2/4, S425 (imágenes disponibles en www.bic.mni.mcgill.ca/brainweb/) (1997).
- [11] D.L. Collins, A.P. Zijdenbos, V. Kollokian, Sled, N.J. Kabian, C.J. Holmes, and A.C. Evans, *Design and construction of a realistic digital brain phantom.*, IEEE Trans. on Med. Imaging (1998), 17(3):463–468.
- [12] C De Castro, E.a.M., *Registration of translated and rotated images using finite fourier transforms*, IEEE Trans. Pattern Analysis and Machine Intelligence (1987), 700–703.
- [13] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern classification*, 2000.
- [14] Rogerio Feris, *Discontinuity preserving stereo with small baseline multi-flash illumination*, International Conference on Computer Vision (2005).
- [15] David J. Fleet and Allan D. Jepson, *Computation of component image velocity from local phase information*, International Journal of Computer Vision (1990), 77–104.
- [16] Stuart Geman and Donald Geman, *Stochastic relaxation, gibbs distributions and the bayes restoration of images*, IEEE Transactions on Pattern Analysis and Machine Intelligence (1986), 721–741.
- [17] D. Greig, B. Porteus, and A. Seheult, *Exact maximum a posteriori estimation for binary images*, Journal of the Royal Statistical Society, Series B (1989), 51(2): 271–279.
- [18] Acosta Héctor, Zitová Barbara, Ríos Homero, Cruz Nicandro, Marín Antonio, Hernández Rodolfo, Cocotle Bertha, and Hernández Efraín, *Cervical cancer detection using colposcopic images: a temporal approach*, Sixth Mexican International Conference on Computer Science (ENC'05) (2005), 158–164.
- [19] Acosta Héctor, Coronel Héctor, Cruz Nicandro, Hernández Rodolfo, and Hernández Efraín, *Modeling aceto-white response functions of precancerous cervical lesions*, Segundo taller de procesamiento de imágenes y óptica (PIO'05) (2005), 158–164.
- [20] IBSR, *Internet brain segmentation repository.*, Technical report, Massachusetts General Hospital, Center for Morphometric Analysis. Available: <http://www.cma.mgh.harvard.edu/ibsr> (2000).

- [21] M Joliot and B. M. Majoyer, *3-d reconstruction of the brain from magnetic resonance images using a connectivity algorithm.*, Magnetic Resonance Imaging (1987), 5:345–352.
- [22] John G. Kemeny and J. Laurie Snell, *Finite markov chains*, 1976.
- [23] V. Kolmogorov and R. Zibih, *What energy functions can be minimized via graph cuts*, ECCV'02 (2002), 65–81.
- [24] R. Leahy, T. Herbert, and R. Lee., *Application of markov random fields in medical imaging*, IPMI (1989), 1–14.
- [25] Brown L.G., *A survey of image registration techniques*, ACM Computing Surveys (1992), 326–376.
- [26] Stan Z. Li, *Markov random field modeling in image analysis*, Springer, New York, 2001.
- [27] J. L. Marroquín, B.C. Vemuri, S. Botello, F. Calderón, and M.D. A. Fernández-Bouzas, *An accurate and efficient bayesian method for automatic segmentation of brain mri*, Comunicación Técnica No. I-02-08/08-05-2002 (CC/CIMAT) (2002).
- [28] J.L. Marroquín, Fernando A. Velasco, Mariano Rivera, and Miguel Nakamura, *Gauss-markov measure field models for low-level vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence (2001).
- [29] Friedman Nir, Geiger D., and G. Moises, *Bayesian network classifiers. machine learning*, 2001.
- [30] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, 1999.
- [31] James R. Norris, *Markov chains*, Cambridge University Press, Cambridge CB2 1TN, UK, 1998.
- [32] J.C. Rajapakse and F. Krugge, *Segmentation of images with intensity inhomogeneities*, Image and Vision Computing (1998), 16:165–180.
- [33] M. Rivera, O. Ocegueda, and J.L. Marroquín, *Entropy controlled gauss-markov random measure field models for early vision*, Variational, Geometric and Level Set Methods in Computer Vision (2005), 137–148.
- [34] Mariano Rivera and James C. Jee, *Twolevel mrf models for image restoration*, Procc. British Machine Vision Conference 2004 (BMVC 2004) (2004), 809–818.

- [35] T.Schiemann, U. Tiede, and K.H. Hohne., *Segmentation of visible human for high quality volume-based visualization*, Medical Image Analysis (1996), 1(4):263–270.
- [36] Mu Zhu and Trevor Hastie, *Feature extraction for nonparametric discriminant analysis*, Journal of Computational and Graphical Statistics (2003), 101–120.