Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Tesis de Licenciatura

Texturas Dinámicas y Segmentación

Juan Manuel Rodríguez

Directora: Marta E. Mejail

Buenos Aires, 23 de Marzo de 2012

Resumen

Las *texturas dinámicas* son secuencias de imágenes de escenas dinámicas que presentan propiedades de regularidad temporal en un sentido estadístico. Incluyen, por ejemplo, olas de mar, humo, torbellinos, fuego o follaje, pero también objetos en movimiento con una "forma definida", como flores o banderas flameando. Este trabajo presenta una caracterización de este tipo de secuencias de video y plantea los problemas de modelado, aprendizaje, síntesis, reconocimiento y segmentación.

Dado que, en ausencia de conocimiento previo adicional, el problema de reconstrucción visual de imágenes es realmente complejo, no se intenta inferir el modelo físico que genera las imágenes. En su lugar se analizan las secuencias de las imágenes únicamente como señales visuales, mediante la construcción de un marco de trabajo estadístico que se basa en disciplinas como el análisis de series de tiempo.

Se propone un modelo estadístico y se expone un método para aprender sus parámetros en el sentido de máxima verosimilitud o de varianza mínima del error de predicción. Se derivan procedimientos de inferencia eficientes en forma cerrada para el aprendizaje de estadísticas de segundo orden. Después de aprender un modelo, éste puede ser utilizado para extrapolar o predecir nueva información de las imágenes, tanto en el dominio temporal como espacial. Se analiza el significado de los parámetros del modelo y se muestra cómo pueden ser manipulados para controlar o animar una simulación.

Utilizando la geometría de los subespacios y la teoría de reconocimiento de patrones, se deriva una técnica para discriminar entre modelos y construir un sistema de reconocimiento. Adicionalmente, mediante la combinación de estos resultados con un marco variacional, se diseña un método de segmentación capaz de dividir una secuencia de video en regiones caracterizadas por diferentes estadísticas espacio-temporales.

Una limitación significativa del modelo de texturas dinámicas es su incapacidad para proporcionar una percepción de descomposición en zonas, donde cada una pertenece a un proceso visual semánticamente diferente. Se observa que, mientras el modelo de texturas dinámicas es inapropiado para este tipo de escenas, el marco generador subyacente sí lo es. De hecho, texturas dinámicas superpuestas pueden ser tenidas en cuenta, aumentando el modelo probabilístico con una variable "oculta" discreta que posea una serie de estados igual al número de texturas y que codifique cuál de ellos es responsable de una determinada pieza espacio-temporal del volumen del video. De esta manera, el volumen de video se puede modelar como una simple textura dinámica, condicionado al estado de la variable oculta. Esto conduce a una extensión natural del modelo de texturas dinámicas, una *mixtura de texturas dinámicas*, o mixtura de sistemas dinámicos lineales, que es estudiada en este trabajo.

La aplicación de una mixtura de texturas dinámicas a la segmentación requiere la descomposición del video en una colección de pequeños parches espacio-temporales, que luego son agrupados. El carácter "local" de esta representación de video es problemática para la segmentación de texturas que son homogéneas globalmente, pero que exhiben una variación sustancial entre ubicaciones vecinas, tales como el movimiento de rotación del agua en un remolino. Además, las segmentaciones basadas en parches tienen una baja precisión en los límites, debido a las fronteras artificiales de los parches subyacentes, y la dificultad de asignar un parche que se superpone en varias regiones de cualquiera de ellos.

Existe también otro modelo, el de *capas de texturas dinámicas*, que representa los videos como una colección de capas, ofreciendo un modelo verdaderamente "global" de la apariencia y la dinámica de cada capa y evita la incertidumbre en los límites. Éste se logra, de manera similar a las mixturas de texturas dinámicas, agregando al modelo original de texturas dinámicas una variable "oculta" discreta que permite la asignación de diferentes comportamientos a diferentes regiones del video. La variable oculta es modelada como un *Campo Aleatorio de Markov* para asegurar la suavidad espacial de las regiones. Condicionada por el estado de esta variable oculta, cada región del video no es más que una textura dinámica. Con la introducción de una representación dinámica compartida para todos los píxeles de una región, el nuevo modelo es una representación en capas.

Además de la introducción de los modelos de mixturas de texturas dinámicas y capas de texturas dinámicas como modelos generativos de video, se derivan los algoritmos de expectation-maximization (EM) para la estimación de máxima verosimilitud de los parámetros de ambos. Se demuestra, asimismo, la aplicabilidad y los inconvenientes que presentan los modelos para la solución de problemas de visión tradicionalmente difíciles, que van desde la agrupación de secuencias de tráfico automovilístico hasta la segmentación de secuencias que contienen múltiples texturas dinámicas.

Índice general

1. Introducción **3**
- 1.1. El problema . . . 3
- 1.2. Las texturas dinámicas como modelo . . . 3
- 1.3. Aplicaciones . . . 4
  - 1.3.1. Edición . . . 4
  - 1.3.2. Compresión . . . 4
  - 1.3.3. Reconocimiento . . . 5
  - 1.3.4. Segmentación . . . 5
- 1.4. Organización . . . 6

2. Texturas dinámicas **7**
- 2.1. El modelo . . . 7
- 2.2. Estimación de los parámetros . . . 8
  - 2.2.1. Solución subóptima . . . 9
- 2.3. Edición . . . 10
  - 2.3.1. Componentes visuales . . . 11
  - 2.3.2. Componentes dinámicas . . . 11
- 2.4. Segmentación . . . 12
  - 2.4.1. Distancia entre texturas dinámicas . . . 13
  - 2.4.2. Segmentación basada en regiones . . . 14

3. Mixturas de texturas dinámicas **17**
- 3.1. El modelo . . . 17
- 3.2. Estimación de los parámetros . . . 19
  - 3.2.1. Etapa E . . . 21
  - 3.2.2. Etapa M . . . 23
- 3.3. Inicialización . . . 25
  - 3.3.1. Semilla inicial . . . 25
  - 3.3.2. Caminos aleatorios . . . 25
  - 3.3.3. División de Componente . . . 25
- 3.4. Clasificación . . . 26
- 3.5. Segmentación . . . 26

4. Capas de texturas dinámicas **28**
- 4.1. El modelo . . . 28
  - 4.1.1. Distribución conjunta . . . 29
- 4.2. Estimación de los parámetros con EM . . . 30
  - 4.2.1. Logaritmo de la verosimilitud completa . . . 31

4.2.2. Etapa E 32
4.2.3. Etapa M 32
4.3. Inferencia aproximada por muestreo de Gibbs 33
4.3.1. Muestrear de $p(Z|X,Y)$ 34
4.3.2. Muestrear de $p(X|Z,Y)$ 35
4.3.3. Inferencia aproximada 36
4.4. Inferencia por aproximación variacional 36
4.4.1. Aproximación de la distribución a posteriori 37
4.4.2. Inferencia aproximada 42
4.5. Inicialización 43
4.6. Segmentación 44

5. Resultados **45**
5.1. Videos con Movimiento circular 46
5.2. Videos de Fenómenos compuestos 47
5.3. Videos Reales 50

6. Conclusiones **53**

Apéndices **55**

A. Conceptos **56**
A.1. Norma Frobenius 56
A.2. Algoritmo EM 56
A.3. Filtro Kalman 57
A.4. Kullback-Leibler 58
A.5. Distancia de Mahalanobis 59

Capítulo 1

Introducción

1.1. El problema

Generalmente, una imagen alcanza para adquirir suficiente conocimiento de una escena para realizar una tarea específica. En otros casos, por el contrario, la descripción semántica deseada del mundo sólo se puede deducir a través del análisis de un grupo de imágenes de la misma escena, tomadas desde diferentes posiciones y/o instantes de tiempo. Por ejemplo, si se necesita determinar si hay un coche estacionado en una esquina, o si hay una bandera en lo alto de un edificio, una imagen de la escena sería suficiente para dar la respuesta correcta. En cambio, si es necesario recuperar la estructura tridimensional del coche o de la bandera (en ausencia de información previa), como mínimo deberíamos disponer de dos imágenes tomadas desde diferentes posiciones, para componer una visión estéreo. Por último, si lo que se necesita es establecer si el coche se mueve con velocidad constante en un intervalo de tiempo determinado, o detectar la presencia de viento y clasificar su intensidad como tranquilo, ligero, moderado o fuerte al mirar la bandera, deberíamos analizar toda una secuencia de imágenes, y esto es porque la información que queremos extraer se codifica no sólo en la forma sino también en la *dinámica*.

En este trabajo estamos interesados en el análisis de secuencias de imágenes de escenas que muestran cambios en el tiempo en su geometría y/o radiometría[1]. Estas secuencias de video tienen ciertas propiedades visuales y dinámicas y las denominamos *procesos visuales dinámicos*.

1.2. Las texturas dinámicas como modelo

Para analizar los procesos visuales dinámicos necesitamos un modelo que los describa. En principio, para entender totalmente las propiedades del proceso visual dinámico, teniendo en cuenta nuestras mediciones (una secuencia finita de imágenes), tenemos que recuperar el modelo físico de la escena que

[1]La radiometría es la medida de radiación óptica electromagnética. Abarca todas las longitudes de onda del espectro electromagnético. Por el contrario, la fotometría sólo se ocupa de la parte visible del espectro, la que puede percibir el ojo humano.

se ha generado. Por desgracia, es bien sabido que la reconstrucción conjunta de la radiometría, la geometría y la dinámica de la escena (problema de la reconstrucción visual) es intrínsecamente un problema mal formulado: a partir de cualquier número (finito) de imágenes no es posible recuperar de forma única todas las incógnitas (forma, pose, reflectancia, distribución de la luz, y movimiento). Esto significa que siempre es posible construir escenas con diferente radiometría, geometría y dinámica que dan origen a las mismas imágenes. Por ejemplo, una secuencia de imágenes del mar al atardecer podría haber sido originada por una forma muy compleja y dinámica (la superficie del mar) con propiedades constantes de reflectancia (material homogéneo, agua) o también por una forma muy sencilla (por ejemplo, una pantalla de televisión), con una dinámica no homogénea de resplandor (la señal espacio-temporal televisada).

Los modelos que se proponen en este trabajo no son modelos de la escena sino modelos estadísticos de la señal de video, es decir, de la secuencia de imágenes en sí. En general, los modelos estadísticos no logran captar correctamente la radiometría, geometría y dinámica de la escena. En su lugar, capturan una mezcla de las tres que es equivalente al modelo físico subyacente de la escena, una vez que el modelo estadístico es "visualizado" como una secuencia de imágenes. Asimismo, la mayor parte del trabajo se dedica a la clase de procesos dinámicos visuales que presentan algún tipo de regularidad temporal. Tales procesos visuales dinámicos incluyen, por ejemplo, las secuencias de fuego, humo, agua, hojas o flores en el viento, las nubes, multitudes de personas que caminan, etc, y nos referiremos a ellos como *texturas dinámicas* [Dor05].

1.3. Aplicaciones

La construcción de modelos basados en video para procesos visuales dinámicos, capaces de predecir o extrapolar nuevas imágenes, tiene un amplio campo de aplicación, que pasamos a analizar.

1.3.1. Edición

Una vez aprendido un modelo generativo, dada una secuencia de imágenes de entrenamiento, se puede recrear la secuencia original. De manera similar, si se modifican los parámetros del modelo y luego se sintetiza, se obtienen nuevas secuencias de video. En el capítulo 2 se describen los algoritmos para llevarlo a cabo.

Las capacidades de los modelos podrían ser utilizadas en las áreas juegos, animación y efectos especiales para cine y televisión. Por ejemplo, se podría animar la foto de una cascada o una bandera en una página web, o bien rellenar un mundo virtual con fuego, nubes, agua, fuentes y banderas, y nunca repetir la mismas imágenes en diferentes momentos y lugares.

1.3.2. Compresión

La compresión de video consiste en la asignación de secuencias de video a cadenas de dígitos binarios. Un codificador de video produce cadenas binarias

cuya longitud es, en promedio, mucho menor que la representación canónica original de las imágenes. Para mejorar el rendimiento de la codificación de las nuevas aplicaciones multimedia, los estándares MPEG-4 y MPEG-7 especifican que una secuencia de video se compone de objetos de video significativos con características homogéneas, como ser intensidad, color, dirección del movimiento u otro tipo de patrón visual predefinido, posiblemente con un significado de mayor nivel semántico. Esta descomposición se puede lograr por medio de análisis de video jerárquico contra un diccionario de clases de modelos. Este marco de trabajo proporciona una contribución útil a muchos aspectos de este problema, como la segmentación, reconocimiento y aprendizaje de modelos simples que pueden ser empleados para representar algunos de los elementos constitutivos de una secuencia de video.

1.3.3. Reconocimiento

Los modelos generativos para los procesos visuales dinámicos se pueden utilizar para realizar tareas de detección y clasificación automática que son aptas, por ejemplo, para vigilancia por video y monitoreo ambiental o de hábitat. Este marco de modelado también puede ser útil para construir sistemas que monitoreen grandes áreas de forma remota mediante la colocación de cámaras en puntos estratégicos, para detectar la presencia de humo o fuego en un bosque o actividad anormal en un estacionamiento o en un piscina o en el medio del océano (por ejemplo, si alguien se está ahogando).

La búsqueda en bases de datos de video (obtención de video) es otra área donde el reconocimiento de la dinámica de los procesos visuales es fundamental. Implica la capacidad de extraer y describir algunos o todos los componentes espaciales y temporales de un video.

1.3.4. Segmentación

La segmentación de una imagen es el proceso de asignación de una etiqueta a cada píxel, de modo que los píxeles que comparten la misma etiqueta también tengan características visuales similares, dando lugar a distintas particiones. El objetivo es simplificar y/o cambiar la representación de la imagen en otra más significativa y más fácil de analizar. Se utiliza tanto para localizar objetos como para encontrar los límites de estos dentro de la imagen. La definición anterior puede ser extendida a secuencias de imágenes, teniendo en cuenta, además de sus características visuales, el comportamiento de los píxeles a lo largo del tiempo.

Los modelos de procesos visuales dinámicos se pueden utilizar para la partición del dominio espacio-temporal de una secuencia de video en regiones, cada una de las cuales se considera como homogénea con respecto a sus propiedades espacio-temporales. Abordar este importante problema no sólo es útil para la obtención de video basado en contenido, sino también para otras tareas como navegación robótica, vigilancia por video, restauración, edición y compresión.

1.4. Organización

El trabajo se encuentra organizado como se describe a continuación. En el capítulo 2 se presenta el método de texturas dinámicas aplicado tanto para la edición como para segmentación de video. En los capítulos 3 y 4 se desarrollan dos extensiones del modelo de texturas dinámicas: las mixturas de texturas dinámicas y las capas de texturas dinámicas. Estas variantes del modelo de texturas dinámicas están orientadas a obtener un rendimiento mayor en lo referido a problemas de segmentación de video. En el capítulo 5 se realizan pruebas utilizando los algoritmos de segmentación de texturas dinámicas, mixturas de texturas dinámicas y capas de texturas dinámicas sobre distintos tipos de videos sintéticos y reales. En el capítulo 6 se exponen las conclusiones y posibles desarrollos futuros. Por último se presenta un anexo A, donde se exponen varios de los algoritmos y métodos que son utilizados en los capítulos. Este sirve como una referencia rápida -pero no exhaustiva- con el objetivo de dar una noción de cada tópico.

Capítulo 2

Texturas dinámicas

En este capítulo se analizan las texturas dinámicas [Dor05], cuál es su modelo y cuáles son las características particulares de ellas. El objetivo es entender cómo "funciona" una textura dinámica y cómo puede ser utilizada para representar un video. Asimismo se presenta un método subóptimo para el aprendizaje de una textura dinámica a partir de un video observado. Adicionalmente, se muestra otra utilidad para las texturas dinámicas, como ser la edición de video.

En la Sección 2.1 se presenta el modelo matemático para representar una textura dinámica y sus parámetros. Luego, en la Sección 2.2, una solución subóptima para aprender los parámetros del modelo a partir de un video. En la Sección 2.3 se muestra cómo pueden ser utilizados los parámetros de la textura dinámica para editar video. Finalmente, en la Sección 2.4, cómo se puede segmentar un video por medio de texturas dinámicas, por medio de segmentación basada en regiones.

2.1. El modelo

Se propone un modelo (ver Figura 2.1) de representación de video basado en dos componentes (dos procesos estocásticos) bien diferenciadas. Una componente, denominada "visual", que controla la apariencia de los cuadros observados, y otra componente, denominada "dinámica", que controla el comportamiento de los cuadros observados. La componente dinámica es representada como un proceso de estados ocultos. Esta componente genera un estado para cada instante de tiempo, que luego utiliza la componente visual para generar un cuadro observado. Se denominan "estados ocultos" a los estados de la componente dinámica porque estos no se pueden observar, a diferencia de los cuadros del video, que sí son observables.

Se define $x_t \in \mathbb{R}^n$ como los estados ocultos, e $y_t \in \mathbb{R}^m$ como los cuadros observados (típicamente $m >> n$), y están relacionados a través del sistema dinámico lineal definido por:

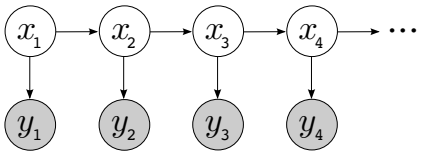

Figura 2.1: Modelo de representación de textura dinámica. La variable x_t representa el estado de la textura dinámica para el instante t, y determina el estado de la variable observada y_t para el mismo instante.

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \tag{2.1}$$

donde $A \in \mathbb{R}^{n\times n}$ es la matriz de transición de estados y $C \in \mathbb{R}^{m\times n}$ es la matriz de observación. Además, las variables aleatorias $v_t \sim \mathcal{N}(0, Q)$ y $w_t \sim \mathcal{N}(0, R)$ representan el ruido percibido, donde $Q \in S_+^n$ y $R \in S_+^m$ (S_+^k el conjunto de matrices k-simétricas definidas-positivas). Generalmente se asume $R = rI_m$, siendo I_m la matriz identidad de orden m y $r \in \mathbb{R}$

Cabe destacar que para generar un estado x_t, se necesita únicamente la matriz de transición de estados A y los estados anteriores $\{x_1, \ldots, x_{t-1}\}$. Esto significa que los estados ocultos x_t "evolucionan" de forma independiente de los cuadros y_t. No así los cuadros y_t, que son determinados por los estados x_t y la matriz de observación C. El instante de tiempo $t \in \mathbb{N}$, varía entre 1 y τ, siendo esta última la cantidad total de estados.

2.2. Estimación de los parámetros

La primera observación con respecto al modelo es que la elección de las matrices A, C, Q, R no es única, en el sentido de que existen infinitas matrices que dan lugar exactamente a los mismas muestras $y(t)$. Esto se puede ver inmediatamente sustituyendo A con TAT^{-1}, C con CT^{-1} y Q con TQT^T, y eligiendo como condición inicial Tx_1, donde $T \in GL(n)$ es cualquier matriz de $n \times n$. En otras palabras, la base del estado-espacio es arbitraria, y un proceso cualquiera no tiene un único modelo, pero sí una clase de equivalencia de modelos $R \doteq \{[A] = TAT^{-1}, [C] = CT^{-1}, [Q] = TQT^T | T \in GL(n)\}$. Para poder identificar un modelo único de una muestra $y(t)$, es necesario elegir un representante de cada clase de equivalencia: cada uno de ellos es llamado una *realización canónica del modelo*, en el sentido de que no depende de la elección de la base del estado-tiempo.

Ya que existen muchas opciones posibles a la hora de definir el modelo, se busca una "a medida" porque es importante que el modelo tenga ciertas propiedades. Una de ellas es la reducción dimensional (utilizar un n) :

$$m >> n;\ \text{rango}(C) = n, \tag{2.2}$$

y otra, que las columnas de C sean ortonormales:

$$C^T C = I_n, \tag{2.3}$$

donde I_n es la matriz identidad de dimensión $n \times n$. Esta propiedad se utiliza luego para obtener los parámetros del modelo.

El problema propuesto puede ser formulado como: dadas las mediciones de una muestra del proceso $y(1), ..., y(\tau)$; $\tau >> n$, estimar $\hat{A}, \hat{C}, \hat{R}, \hat{Q}$, un modelo canónico del proceso $\{y(t)\}$. Idealmente, se busca la solución de máxima-verosimilitud:

$$\hat{A}(\tau), \hat{C}(\tau), \hat{Q}(\tau), \hat{R}(\tau) = \arg \max_{A,C,Q,R} p(y(1), ..., y(\tau)) \tag{2.4}$$

En la literatura de teoría de identificación de sistemas [Lju99] existen soluciones óptimas de carácter asintótico para este problema, en el sentido de máxima verosimilitud. Pero dada la dimensión de los problemas que se tratan (los píxeles de una serie de imágenes que forman el video) estas soluciones no son aptas por el tiempo de respuesta necesario para obtener los resultados. Es por ello que se propone una solución subóptima que respeta las propiedades anteriormente expuestas.

2.2.1. Solución subóptima

La solución subóptima hallada utiliza la descomposición en valores singulares o SVD [GVL96], que es una manera eficiente, aunque no exacta, de obtener los parámetros del sistema que cumplen con los requisitos.

Sean

$$Y_1^\tau \doteq [y(1), ..., y(\tau)] \in \mathbb{R}^{m \times \tau} \text{ una observación} \tag{2.5}$$

$$X_1^\tau \doteq [x(1), ..., x(\tau)] \in \mathbb{R}^{n \times \tau} \text{ una secuencia de estados} \tag{2.6}$$

Se busca que

$$Y_1^\tau = C X_1^\tau \tag{2.7}$$

donde $C \in \mathbb{R}^{m \times n}$ cumpliendo $C^T C = I$, por lo asumido en (2.3).

Se define

$$Y_1^\tau = U \Sigma V^T \tag{2.8}$$

como la descomposición SVD donde $U \in \mathbb{R}^{m \times n}$ $(U^T U = I_n)$, $V \in \mathbb{R}^{\tau \times n}$ $(V^T V = I_n)$ y $\Sigma = \text{diag}\{\sigma_1, ...\sigma_n\}$ con los valores singulares $\{\sigma_i\}$. Considerar el problema de encontrar la mejor estimación de C, en el sentido de la norma Frobenius (A.1):

$$\hat{C}(\tau), \hat{X}(t) = \arg \min_{C, X_1^\tau} \left\| Y_1^\tau - C X_1^\tau \right\|_F . \tag{2.9}$$

Se puede observar inmediatamente que la única solución, por la propiedad SVD de aproximación de rango fijo, está dada por:

$$\hat{C}(\tau) = U, \ \hat{X}(\tau) = \Sigma V^T. \tag{2.10}$$

De manera similar, $\hat{A}$ puede ser determinada de forma única, en el sentido de Frobenius, resolviendo:

$$\hat{A} = \arg\min_{A} \left\| X_2^{\tau} - A X_1^{\tau-1} \right\|_F , \tag{2.11}$$

donde $X_1^{\tau} \doteq [x(1), ..., x(\tau)] \in \mathbb{R}^{n \times \tau}$ que se resuelve utilizando (2.12):

$$\hat{A}(\tau) = \hat{X}_2^{\tau} (\hat{X}_1^{\tau-1})^{-1}, \tag{2.12}$$

Tener en cuenta que lo que se calcula en 2.12 es la pseudo-inversa, ya que la matriz $\hat{X}_1^{\tau-1}$ no es necesariamente cuadrada.

Finalmente, se calculan las matrices de covarianza R (ruido de entrada de la variable observada) y Q (ruido de entrada de la variable oculta), que pueden ser estimadas como:

$$\hat{R}(\tau) = \frac{1}{\tau} \sum_{i=1}^{\tau} \hat{w}(i) \hat{w}(i)^T \tag{2.13}$$

$$\hat{Q}(\tau) = \frac{1}{\tau} \sum_{i=1}^{\tau} \hat{v}(i) \hat{v}(i)^T, \tag{2.14}$$

donde

$$\hat{w}(t) \doteq y(t) - \hat{C}(\tau) \hat{x}(t) \tag{2.15}$$

$$\hat{v}(t) \doteq \hat{x}(t+1) - \hat{A}(\tau) \hat{x}(t). \tag{2.16}$$

En todo momento se asume que el orden del modelo está determinado por un n dado. En la práctica, esto tiene que ser inferido. Una manera empírica para calcular el orden del modelo es a través de sus valores singulares, descartando aquellos que estén por debajo de un umbral. Como umbral se puede utilizar una constante o una cota, como la diferencia entre dos valores singulares adyacentes.

2.3. Edición

Si bien no es el objetivo del trabajo, la edición de video a través de texturas dinámicas es de gran utilidad para comprender mejor las propiedades del proceso visual. Esto explica cómo se conectan la estructura y la intensidad de los parámetros del modelo a la apariencia del proceso visual. Un ejemplo de los resultados que se pueden obtener por medio de la edición de video con texturas dinámicas se puede apreciar en la Figura 2.2.

Hasta el momento, sabemos que el procedimiento de aprendizaje de la Sección 2.2 produce las matrices $\hat{A}, \hat{C}, \hat{Q}, \hat{R}$ que se pueden utilizar para sintetizar y generar un video $\{\hat{I}(t)\}$. En principio puede resultar obvio que, si se realiza cualquier cambio sobre los parámetros, se obtenga como resultado un nuevo video, distinto del original. Pero la modificación arbitraria raramente genera una secuencia de imágenes similares a un fenómeno real. De hecho, $\hat{A}$ debe ser estable (autovalores dentro del círculo complejo unitario) y $\hat{C}$ debe tener columnas ortogonales (para obtener una realización canónica). Las matrices $\hat{Q}, \hat{R}$ no se analizan (aunque como se ha mencionado introducen cambios en la secuencia resultante) ya que representan los ruidos de las componentes visual y dinámica correspondientemente.

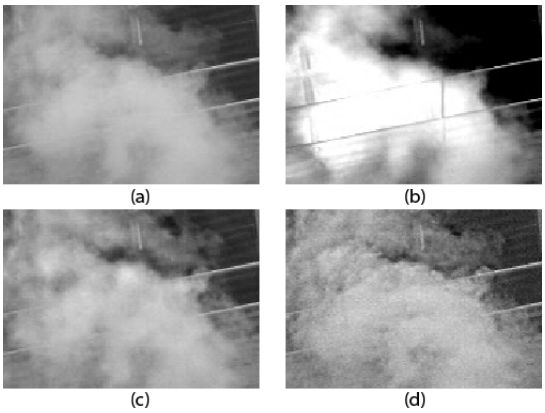

Figura 2.2: Resultados de la edición de un video con humo. Se muestra el mismo instante de tiempo t de cuatro videos. El primero de ellos es el original (a), y los restantes fueron sintetizados a partir del original, duplicando el valor de las escalas gruesas (b), medias (c) y finas (d) respectivamente.

2.3.1. Componentes visuales

La matriz $\hat{C}$ tiene en sus columnas las primeras n componentes principales. Estas componentes son, por construcción, una base ortonormal que cubre un subespacio en $\mathbb{R}^m$. Además, están ordenadas de la primera a la última columna, de forma que se incrementa la frecuencia espacial que representan. Por lo tanto, las primeras componentes (las primeras columnas de C) representan las escalas espaciales gruesas del patrón de la textura y las últimas (las últimas columnas de C) representan las más finas.

Teniendo en cuenta lo anterior podemos realizar cierto tipo de manipulación. Se puede deformar el subespacio cubierto por las componentes principales cambiando los "pesos" a cada una de ellas. Es decir, esto se hace substituyendo la matriz $\hat{C}$ por otra matriz $\tilde{C} \doteq \hat{C}W$, siendo $W \in \mathbb{R}^{n \times n}$ una matriz diagonal con $w_1 \dots w_n$ números reales positivos como elementos de su diagonal. Luego, lo único que resta es sintetizar utilizando la matriz modificada.

2.3.2. Componentes dinámicas

Intuitivamente, si se quisiera duplicar la velocidad de un video se podría reemplazar la matriz de transición A por $\tilde{A} \doteq AA$, ya que de esta manera al realizar un cambio de estado se obtendría $AAx(t) = Ax(t+1) = x(t+2)$ (asumiendo que no hay ruido). Más allá de que no es la solución exacta, es una buena aproximación.

Para alterar la velocidad adecuadamente, primero se tiene que descomponer $A = V\Lambda V^{-1}$, donde Λ y V son respectivamente las matrices de autovalores y autovectores. Luego, se pasa Λ a coordenadas polares $\{|\lambda_i|\exp(\psi_i)\}_{i=1\dots n}$. Multiplicando cada ángulo por una constante de velocidad Ω, se obtiene el

efecto buscado:

$$\hat{\psi}_i \doteq \Omega\psi_i \tag{2.17}$$

Por ejemplo, queremos duplicar la velocidad, entonces utilizamos como constante de velocidad $\Omega = 2$

El motivo por el cual utilizar potencias no funciona, es que la potencia A^{Ω}, además de modificar los ángulos $\Omega\psi_i$, modifica también las distancias $|\lambda_i|^{\Omega}$, lo que no es correcto salvo que todas las distancias sean 1.

2.4. Segmentación

En esta Sección se estudia el problema de la segmentación de una secuencia de imágenes basado en el modelo de la Sección 2.1.

La segmentación de un video consiste en particionar el dominio de una imagen, en dos o más regiones disjuntas, donde las regiones no varían en el tiempo. Se espera que los elementos presentes en una región estén relacionados entre sí, es decir, que tengan un comportamiento similar. O sea, cada región debe representar un fenómeno diferente, ya sea, por su dinámica y/o por su apariencia.

El método de segmentación que se propone, utiliza un algoritmo de segmentación de imágenes. El método de segmentación de imágenes agrupan los píxeles de acuerdo a su intensidad. En este caso, en vez de utilizar la intensidad del píxel, se utiliza la distancia (definida en la Subsección 2.4.1) entre el modelo del píxel y un modelo de referencia. En realidad, una vez calculadas las distancias se puede utilizar cualquier técnica de segmentación de imágenes.

Antes de proceder con la formalización del problema, cabe señalar que la segmentación es totalmente dependiente de la clase de los modelos elegidos. Diferentes modelos resultan en diferentes particiones de la escena, y no hay un resultado correcto o incorrecto. En última instancia, la utilidad del método de segmentación depende de qué tan bien el modelo elegido captura la "esencia" de la escena. Esto significa que, si no se tiene un modelo preciso para empezar, esta correspondencia no puede ser garantizada.

Los modelos de referencia pueden ser conocidos, cuando se sabe lo que se espera encontrar en la escena (fuego, agua, etc), o no. En el primer caso sería sencillo segmentar, ya que se calculan las distancias a los píxeles y luego se agrupan utilizando la distancia mínima. El segundo es más complejo, ya que ni siquiera se sabe cuántas clases están presentes en la escena. En nuestro trabajo, se asume que los modelos de referencia son conocido, como lo es para un sistema supervisado.

2.4.1. Distancia entre texturas dinámicas

Para calcular la distancia entre dos texturas dinámicas se utilizan únicamente los parámetros $\{A, C\}$, ya que se considera que dos procesos con diferente ruido son equivalentes.

Como función de distancia se podría utilizar simplemente la norma Frobenius, pero los resultados que se obtienen tomando esta medida son bastante pobres. Debido a esto, se utilizan los ángulos principales que se forman entre los subespacios de los modelos de las texturas dinámicas.

Sean S_A y S_B dos subespacios de dimensión p y q $(p \geq q)$, respectivamente, y sean $A \in \mathbb{R}^{m\times p}$ y $B \in \mathbb{R}^{m\times q}$ sus representaciones matriciales. Definimos los ángulos principales entre los subespacios S_A y S_B, $\theta_k \in [0, \frac{\pi}{2}]; k = \{1, ..., q\}$, de la siguiente manera:

$$\begin{aligned}\cos\theta_k &= \max_{\substack{x\in\mathbb{R}^p \\ y\in\mathbb{R}^q}} \frac{|x^T A^T B y|}{\|Ax\|_2 \|By\|_2} \\ &= \frac{|x_k^T A^T B y_k|}{\|Ax_k\|_2 \|By_k\|_2}, \text{ para } k = 1, ..., q\end{aligned}$$

sujeto a:

$$\begin{aligned}&x_k \neq 0, y_k \neq 0, \\ &x_i^T A^T A x_k = y_i^T B^T B y_k = 0, \text{ para } i = 1, ..., k-1,\end{aligned}$$

donde las matrices A y B tienen rango completo (linealmente independientes).

La definición anterior nos sirve para calcular la distancia entre dos modelos, $M_1 \doteq (A_1, C_1)$ y $M_2 \doteq (A_2, C_2)$, por medio de sus matrices de observabilidad infinita. La matriz de observabilidad infinita se define como:

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \doteq \mathcal{O} \in \mathbb{R}^{\infty\times n} \tag{2.18}$$

y es una forma de describir tanto la dinámica como la apariencia del modelo en una sola matriz, ya que:

$$\begin{bmatrix} y(1) \\ y(2) \\ y(3) \\ \vdots \end{bmatrix} \approx \begin{bmatrix} Cx(1) \\ Cx(2) \\ Cx(3) \\ \vdots \end{bmatrix} \approx \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} x(1) = \mathcal{O}\, x(1)$$

Dado que el modelo se genera a partir de una entrada finita $Y \in \mathbb{R}^{m\times\tau}$, el tamaño de la matriz de observabilidad es $\mathcal{O} \in \mathbb{R}^{m\tau\times n}$, donde m es la cantidad de filas por columnas de los cuadros, y τ es la cantidad de cuadros del video de entrada. Esto no afecta la cantidad de ángulos, ya que estos dependen del rango de las matrices, que sigue siendo n.

Los ángulos principales entre los subespacios $\mathcal{O}_1$ y $\mathcal{O}_2$ son obtenidos resolviendo el siguiente problema de autovalores generalizado [CCMM00]:

$$\begin{bmatrix} 0 & A^TB \\ B^TA & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} A^TA & 0 \\ 0 & B^TB \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

sujeto a:

$$x^TA^TAx = y^TB^TBy = 1.$$

Al resolver este sistema se obtienen $2n$ autovalores reales. Si se asume que están ordenados en forma descendente:

$$\lambda_1 \geq ... \geq \lambda_{2n}$$

se puede demostrar que:

$$\lambda_1 = \cos\theta_1, ..., \lambda_n = \cos\theta_n \geq 0,$$
$$\lambda_{2n} = -\cos\theta_1, ..., \lambda_{n+1} = -\cos\theta_n$$

Finalmente, se utiliza como distancia entre dos modelos de texturas dinámicas M_1 y M_2, la distancia de *Martin* [Mar00], definida de la siguiente manera:

$$d_M^2(M_1, M_1) = \ln \prod_{i=1}^{n} \frac{1}{\cos^2\theta_i}. \tag{2.19}$$

La distancia de *Martin* calcula la distancia entre los modelos M_1 y M_1, utilizando los ángulos principales que forman los subespacios de sus matrices de observabilidad, $\mathcal{O}_1$ y $\mathcal{O}_1$.

También es posible utilizar la distancia de *Finsler* [Wei00], que utiliza sólamente el más grande de los ángulos:

$$d_F = \theta_1, \tag{2.20}$$

pero esta última muestra resultados más pobres.

2.4.2. Segmentación basada en regiones

Ahora que se ha definido la función de distancia (2.19) entre dos texturas dinámicas, se puede proceder a detallar el procedimiento para llevar a cabo la segmentación de un video que presente, al menos, dos fenómenos relevantes.

El método en cuestión primero genera una imagen a partir del video, que luego es segmentada utilizando una técnica de segmentación para imágenes. Esta imagen codifica en cada uno de sus píxeles la apariencia y dinámica de cada posición del video a lo largo del tiempo, y es por esto que el resultado de la segmentación sobre esta imagen resuelve la segmentación sobre el video. Ahora se procede a detallar el algoritmo.

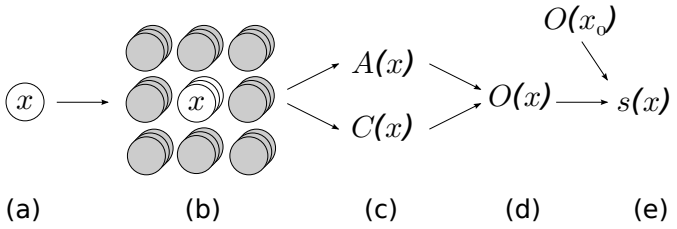

Figura 2.3: Cálculo de la firma para un píxel x. (a) Dado un píxel x de la imagen. (b) Se toma una vecindad espacio-temporal alrededor de x. (c) Se aprenden los parámetros del modelo $A(x)$ y $C(x)$. (d) Se calcula la matriz de observabilidad $\mathcal{O}(x)$ (2.18). (e) Utilizando el modelo de referencia $\mathcal{O}(x_0)$, se calcula la distancia a $\mathcal{O}(x)$, obteniendo la firma para x.

Sea $\Omega \subset \mathbb{R}^2$ el dominio de una imagen y $\{\Omega_i\}_{i=1\ldots N}$ una partición de Ω en N regiones disjuntas:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \text{ y } \Omega_i \cap \Omega_j = \emptyset \text{ para } i \neq j \tag{2.21}$$

Se calcula la "firma" espacio-temporal (ver Figura 2.3) para cada píxel $x \in \Omega$. Esto se realiza tomando la vecindad $B(x) \subset \Omega$ alrededor de x, aprendiendo los parámetros $A(x)$ y $C(x)$ y ejecutando el procedimiento de la Sección 2.2. Con estos se genera $\mathcal{O}(x)$ por medio de la ecuación (2.18). Finalmente, la firma[1] para cada píxel x es obtenida calculando la distancia (2.19) entre el modelo $\mathcal{O}(x)$ y un modelo de referencia $\mathcal{O}(x_0)$

$$s(x) = d_M^2(\mathcal{O}(x), \mathcal{O}(x_0)) = \ln \prod_{i=1}^{n} \frac{1}{\cos^2 \theta_i(x)}. \tag{2.22}$$

La segmentación de Ω en regiones disjuntas Ω_i con firmas constantes $s_i \in \mathbb{R}^n$ se obtiene minimizando la función de costo

$$E(\Gamma, \{s_i\}) = \sum_{i=1}^{N} \int_{\Omega_i} (s(x) - s_i)^2 dx + v|\Gamma| \tag{2.23}$$

simultáneamente con respecto a los descriptores de la región $\{s_i\}$, que modelan el valor medio de firma para cada región, y con respecto al borde Γ, que separa las regiones. La primera componente busca que los elementos de la región sean similares (apariencia y comportamiento), mientras la segunda, que el borde de la región sea mínimo. v se utiliza para indicar qué tan importante es el tamaño del borde.

De ahora en adelante, todas las clases de soluciones se restringen al caso particular de segmentación de dos regiones, aunque se puede extender para tres

[1]Como firma también se podría utilizar un vector formado por los ángulos entre los subespacios de los modelos, o cualquier otra medida que nos indique que tan similares son dos modelos entre sí.

o más regiones.

Para la implementación del borde Γ se utiliza la representación basada en conjuntos de nivel implícito [CSV00]. Entonces se define el borde Γ como el conjunto de nivel cero de una función $\phi : \Omega \to \mathbb{R}$:

$$\Gamma = \{x \in \Omega | \phi(x) = 0\}, \tag{2.24}$$

La motivación de ϕ es dar información acerca de las regiones. En los lugares donde $\phi > 0$ se encuentran los píxeles que pertenecen a una de las regiones, mientras que en $\phi < 0$ la otra. Y cuando $\phi = 0$, se presenta el borde.

Ahora se puede ver como queda la función de costo utilizando conjuntos de nivel:

$$\begin{aligned} E(\phi, \{s_i\}) = & \int_{\Omega} (s(x) - s_1)^2 H(\phi) dx \\ & + \int_{\Omega} (s(x) - s_2)^2 (1 - H(\phi)) dx \\ & + v|\Gamma| \end{aligned} \tag{2.25}$$

donde

$$H(\phi) = \begin{cases} 1 & \text{si } \phi \geq 0 \\ 0 & \text{si } \phi < 0 \end{cases} \tag{2.26}$$

Se puede observar como $H(\phi)$ anula los píxeles que están fuera de la región a la que pertenecen. Esto es necesario ya que las integrales abarcan todo Ω.

Ahora, se derivan cada una de las variables de la función de costo, para hallar los mínimos. Esto requiere de dos etapas:

- Estimación de la firma media.
  Para un ϕ fijo, se minimiza con respecto a las firmas $\{s_i\}$ de las regiones, lo que equivale a calcular la firma media de los píxeles de cada región:

$$s_1 = \frac{\int sH(\phi)dx}{\int H(\phi)dx}, \; s_2 = \frac{\int s(1 - H(\phi))dx}{\int (1 - H(\phi))dx} \tag{2.27}$$

- Evolución de los bordes.
  Para $\{s_i\}$ fijos, la minimización de la función ϕ puede ser llevar a cabo por medio de un descenso de gradiente dado por:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[v \nabla \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + (s - s_2)^2 - (s - s_1)^2 \right] \tag{2.28}$$

Iterando entre estas dos etapas se obtienen las regiones que minimizan la función de costo, lo que conduce al resultado buscado, la segmentación de video en regiones disjuntas.

En el Capítulo 5 se pueden observar los resultados del algoritmo de segmentación basado en regiones aplicado a distintas clases de videos.

Capítulo 3

Mixturas de texturas dinámicas

3.1. El modelo

Como se ha visto en el Capítulo 2, una textura dinámica es un modelo generativo tanto para la apariencia como para la dinámica de un video. Está formado por un proceso aleatorio conteniendo una variable observada y_t, que representa la apariencia (cuadro de video en el instante t), y una variable de estado oculta x_t, que representa la dinámica (evolución del video en el tiempo). Las variables observadas y de estado están relacionadas a través del Sistema Dinámico Lineal (SDL) definido por:

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \tag{3.1}$$

donde $x_t \in \mathbb{R}^n$ e $y_t \in \mathbb{R}^m$. El parámetro $A \in \mathbb{R}^{n \times n}$ es una matriz de transición de estado y $C \in \mathbb{R}^{m \times n}$ es una matriz de observación. Los procesos generadores de ruido v_t y w_t tienen distribución normal con media cero y covarianza Q y R, respectivamente, es decir, $v_t \sim \mathcal{N}(0, Q)$ y $w_t \sim \mathcal{N}(0, R)$. A este modelo se lo extiende para que soporte un estado inicial x_1 de media μ y covarianza arbitraria S, o sea, $x_1 \sim \mathcal{N}(\mu, S)$. Esto permite capturar variabilidad en el primer cuadro. Por lo tanto, una textura dinámica queda definida por $\Theta = \{A, Q, C, R, \mu, S\}$.

De esta definición se desprende que las distribuciones del estado inicial, del estado condicional y de la observación condicional son:

$$p(x_1) = G(x_1, \mu, S), \tag{3.2}$$
$$p(x_t|x_{t-1}) = G(x_t, Ax_{t-1}, Q), \tag{3.3}$$
$$p(y_t|x_t) = G(y_t, Cx_t, R), \tag{3.4}$$

donde $G(x, \mu, \Sigma) = (2\pi)^{-n/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}\|x-\mu\|^2_\Sigma}$ es una distribución gaussiana n-dimensional, y $\|x\|^2_\Sigma = x^T\Sigma^{-1}x$ es la distancia Mahalanobis con respecto a la

matriz de covarianza Σ.

Habiendo definido lo anterior, y dado que $x_1^\tau = (x_1, \cdots, x_\tau)^T$ es la secuencia de estados e $y_1^\tau = (y_1, \cdots, y_\tau)^T$ es la secuencia de observaciones (cuadros de video), la función de distribución conjunta es:

$$p(x_1^\tau, y_1^\tau) = p(x_1) \prod_{t=2}^{\tau} p(x_t|x_{t-1}) \prod_{t=1}^{\tau} p(y_t|x_t) \tag{3.5}$$

Bajo el modelo de mixturas de texturas dinámicas [CV08] (ver Figura 3.1), el video observado proviene de una de K texturas dinámicas, donde cada una de éstas tiene probabilidad mayor a cero de ocurrencia. Esta es una extensión útil para dos clases de aplicaciones. La primera involucra video que es homogéneo a cada instante, pero que tiene estadísticas que cambian a lo largo del tiempo. Por ejemplo, el problema de agrupar un conjunto de videos tomados de una cámara fija en una autopista. Más allá que el video muestra tráfico moviéndose a velocidad homogénea, la apariencia exacta de cada secuencia es controlada por la cantidad de congestión. Diferentes grados de congestión pueden ser representados por diferentes texturas dinámicas. La segunda involucra video no homogéneo, esto es, compuesto de múltiples procesos que pueden ser individualmente modelados como texturas dinámicas con diferentes parámetros. Por ejemplo, en una escena que contiene fuego y humo, un parche de video tomado al azar del video contendrá fuego o humo, y una colección de parches de video puede ser representada como una muestra de una mixtura de dos texturas dinámicas.

Al modelo de texturas dinámicas se agrega la variable $z \sim multinomial(\alpha_1, ..., \alpha_K)$ con $\sum_{j=1}^{K} \alpha_j = 1$. Asimismo, ahora tenemos K parámetros de componentes de texturas dinámicas $\{\Theta_1, \ldots, \Theta_K\}$. La finalidad de z es indicar cuál de las K texturas dinámicas se utiliza para representar a la secuencia observada y_t.

La probabilidad de una secuencia observada y_t bajo este modelo es:

$$p(y_1^\tau) = \sum_{j=1}^{K} \alpha_j p(y_1^\tau|z = j), \tag{3.6}$$

donde $p(y_1^\tau|z = j)$ es la probabilidad del video, dado que se utiliza la j-ésima textura dinámica.

Ahora, el sistema de ecuaciones de la mixtura de texturas dinámicas es:

$$\begin{cases} x_{t+1} = A_z x_t + v_t \\ y_t = C_z x_t + w_t \end{cases} \tag{3.7}$$

donde la variable aleatoria z indica la componente de textura de donde provienen las observaciones (el video). La condición inicial está dada por $x_1 \sim \mathcal{N}(\mu_z, S_z)$ y los ruidos por $v_t \sim \mathcal{N}(0, Q_z)$ y $w_t \sim \mathcal{N}(0, R_z)$. Las distribuciones condicionales

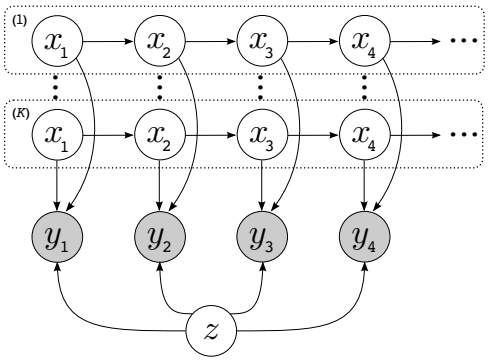

Figura 3.1: Las mixturas de texturas dinámicas. La variable observada y es el proceso de píxeles observados, mientras que x es el proceso de estados ocultos. Existen K procesos ocultos donde la variable oculta z_i indica de qué componente de textura dinámica proviene la muestra de video (y_i).

de los estados y las observaciones, dado el índice de la componente z, son:

$$p(x_1|z) = G(x_1, \mu_z, S_z), \tag{3.8}$$
$$p(x_t|x_{t-1}, z) = G(x_t, A_z x_{t-1}, Q_z), \tag{3.9}$$
$$p(y_t|x_t, z) = G(y_t, C_z x_t, R_z), \tag{3.10}$$

y la función de distribución conjunta es:

$$p(x_1^\tau, y_1^\tau, z) = p(z)p(x_1|z)\prod_{t=2}^{\tau} p(x_t|x_{t-1}, z)\prod_{t=1}^{\tau} p(y_t|x_t, z) \tag{3.11}$$

3.2. Estimación de los parámetros

Estimar los parámetros del modelo utilizando como entrada sola una secuencia de video observada no tiene sentido, ya que para realizar esto solo es necesaria una sola textura dinámica. En realidad, lo que se busca es hallar los parámetros de K texturas dinámicas que mejor se "ajusten" a un conjunto de N secuencias de video, donde generalmente $N >> K$.

Por lo tanto, dado un conjunto de N variables aleatorias i.i.d. (secuencias de video) $\{y^{(i)}\}_{i=1}^N$, nos gustaría aprender los parámetros $\Theta = \{A, Q, C, R, \mu, S\}$ de una mixtura de texturas dinámicas que mejor se ajustan a la información, en el sentido de máxima verosimilitud, esto es:

$$\Theta^* = \arg\max_{\Theta} \sum_{i=1}^{N} \log p(y^{(i)}; \Theta) \tag{3.12}$$

Cuando la probabilidad de la distribución depende de las variables ocultas, la solución de máxima verosimilitud puede ser encontrada con el algoritmo EM (ver Apéndice A.2). El algoritmo EM es un método iterativo para hallar la información faltante (variables ocultas y parámetros del modelo) por medio de la

información observada (variables observadas). Este algoritmo alterna entre una etapa donde se calculan las esperanzas de las variable ocultas, y otra donde se calculan los parámetros del modelo.

Para la mixtura de texturas dinámicas, la información observada es el conjunto de secuencias de video $\{y^{(i)}\}_{i=1}^{N}$, y la información que falta es:

1. La asignación $z^{(i)}$ de cada secuencia a una componente de mixtura.
2. La secuencia de estados ocultos $x^{(i)}$ que genera $y^{(i)}$.
3. Los parámetros Θ de las K componentes de texturas dinámicas.

El algoritmo EM para mixturas de texturas dinámicas, en cada iteración calcula:

$$\text{Etapa E:} \quad \mathcal{Q}(\Theta;\hat{\Theta}) = \mathbb{E}_{X,Z|Y;\hat{\Theta}}(\log p(X,Y,Z;\Theta)) \tag{3.13}$$

$$\text{Etapa M:} \quad \hat{\Theta}^{*} = \arg\max_{\Theta} \mathcal{Q}(\Theta;\hat{\Theta}) \tag{3.14}$$

donde $p(X,Y,Z;\Theta)$ es la probabilidad de todas las observaciones, estados ocultos y asignación de variables ocultas, parametrizadas por Θ. Se asume que las observaciones son independientes y tienen media cero, aunque el algoritmo puede ser extendido para el caso de medias distintas de cero.

En (3.15) se define el logaritmo de la verosimilitud de toda la información, que es utilizado en las etapas E y M, para obtener los parámetros de la distribución.

Se introduce el vector $z_i \in \{0,1\}^K$, tal que $z_{i,j} = 1$ si y sólo si $z^{(i)} = j$, como es común en EM [DLR77]. Su finalidad es anular los términos de la función que no pertenecen a una componente de la mixtura dada.

$$\ell(X,Y,Z) = \sum_{i=1}^{N} \log p(x^{(i)}, y^{(i)}, z^{(i)}) \tag{3.15}$$

$$= \sum_{i=1}^{N} \log \prod_{j=1}^{K} \left[p\left(x^{(i)}, y^{(i)}, z^{(i)} = j\right) \right]^{z_{i,j}} \tag{3.16}$$

$$= \sum_{i,j} z_{i,j} \log \left[\alpha_j p\left(x_1^{(i)} | z^{(i)} = j\right) \right. \tag{3.17}$$

$$\left. \cdot \prod_{t=2}^{\tau} p\left(x_t^{(i)} | x_{t-1}^{(i)}, z^{(i)} = j\right) \prod_{t=1}^{\tau} p\left(y_t^{(i)} | x_t^{(i)}, z^{(i)} = j\right) \right]$$

$$= \sum_{i,j} z_{i,j} \left[\log \alpha_j + \log p\left(x_1^{(i)} | z^{(i)} = j\right) \right. \tag{3.18}$$

$$\left. + \sum_{t=2}^{\tau} \log p\left(x_t^{(i)} | x_{t-1}^{(i)}, z^{(i)} = j\right) + \sum_{t=1}^{\tau} \log p\left(y_t^{(i)} | x_t^{(i)}, z^{(i)} = j\right) \right]$$

Si se observan las funciones de probabilidad de (3.8)-(3.10), las sumas de los

logaritmos de la probabilidades condicionales son de la forma:

$$\sum_{i,j} a_{i,j} \sum_{t=t_0}^{t_1} \log G(b_t, c_{j,t}, M_j) = -\frac{d}{2}(t_1 - t_0 + 1) \log 2\pi \sum_{i,j} a_{i,j} \tag{3.19}$$
$$- \frac{1}{2} \sum_{i,j} a_{i,j} \sum_{t=t_0}^{t_1} \|b_t - c_{j,t}\|^2_{M_j} - \frac{t_1 - t_0 + 1}{2} \sum_{i,j} a_{i,j} \log |M_j|$$

Dado que el primer término no depende de los parámetros de la mixtura de texturas dinámicas, no tiene efecto en la maximización de la etapa M y es por eso que puede ser descartado. Sustituyendo cada una de las funciones de probabilidad gaussiana, se obtiene:

$$\ell(X, Y, Z) = \sum_{i,j} z_{i,j} \log \alpha_j \tag{3.20}$$
$$- \frac{1}{2} \sum_{i,j} z_{i,j} \log |S_j| - \frac{1}{2} \sum_{i,j} z_{i,j} \|x_1^{(i)} - \mu\|^2_{S_j}$$
$$- \frac{\tau - 1}{2} \sum_{i,j} z_{i,j} \log |Q_j| - \frac{1}{2} \sum_{i,j} z_{i,j} \sum_{t=2}^{\tau} \|x_t^{(i)} - A_j x_{t-1}^{(i)}\|^2_{Q_j}$$
$$- \frac{\tau}{2} \sum_{i,j} z_{i,j} \log |R_j| - \frac{1}{2} \sum_{i,j} z_{i,j} \sum_{t=1}^{\tau} \|y_t^{(i)} - C_j x_t^{(i)}\|^2_{R_j}$$

y luego, ya que $\|x\|^2_\Sigma = x^T \Sigma^{-1} x = \mathrm{tr}(x^T \Sigma^{-1} x) = \mathrm{tr}[\Sigma^{-1}(xx^T)]$:

$$\ell(X, Y, Z) = \sum_{i,j} z_{i,j} \log \alpha_j \tag{3.21}$$
$$- \frac{1}{2} \sum_{i,j} z_{i,j} \log |S_j|$$
$$- \frac{1}{2} \sum_{i,j} z_{i,j} \mathrm{tr} \left[S_j^{-1} \left(P_{1,1}^{(i)} - x_1^{(i)} \mu_j^T - \mu_j {x_1^{(i)}}^T + \mu_j \mu_j^T \right) \right]$$
$$- \frac{\tau}{2} \sum_{i,j} z_{i,j} \log |R_j|$$
$$- \frac{1}{2} \sum_{i,j} z_{i,j} \sum_{t=1}^{\tau} \mathrm{tr} \left[R_j^{-1} \left(y_t^{(i)} {y_t}^{(i)T} - y_t^{(i)} {x_t}^{(i)T} C_j^T - C_j P_{t,t}^{(i)} C_j^T \right) \right]$$
$$- \frac{\tau - 1}{2} \sum_{i,j} z_{i,j} \log |Q_j|$$
$$- \frac{1}{2} \sum_{i,j} z_{i,j} \sum_{t=2}^{\tau} \mathrm{tr} \left[Q_j^{-1} \left(P_{t,t}^{(i)} - P_{t,t-1}^{(i)} A_j^T - A_j {P_{t,t-1}^{(i)}}^T + A_j P_{t-1,t-1}^{(i)} A_j^T \right) \right]$$

donde $P_{t,t}^{(i)} = x_t^{(i)} (x_t^{(i)})^T$ y $P_{t,t-1}^{(i)} = x_t^{(i)} (x_{t-1}^{(i)})^T$.

3.2.1. Etapa E

La etapa E del algoritmo EM consiste en tomar la esperanza de (3.21) condicionada por la información observada y los parámetros estimados actuales $\hat{\Theta}$.

Se puede observar que cada término de $\ell(X, Y, Z)$ es de la forma $z_{i,j} f(x^{(i)}, y^{(i)})$, para una función f de $x^{(i)}$ y $y^{(i)}$, y su esperanza es:

$$\mathbb{E}_{X,Z|Y}\left(z_{i,j} f(x^{(i)}, y^{(i)})\right) \tag{3.22}$$

$$= \mathbb{E}_{Z|Y}\left(\mathbb{E}_{X|Y,Z}\left(z_{i,j} f(x^{(i)}, y^{(i)})\right)\right) \tag{3.23}$$

$$= \mathbb{E}_{z^{(i)}|y^{(i)}}\left(\mathbb{E}_{x^{(i)}|y^{(i)},z^{(i)}}\left(z_{i,j} f(x^{(i)}, y^{(i)})\right)\right) \tag{3.24}$$

$$= p(z_{i,j} = 1|y^{(i)})\mathbb{E}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(f(x^{(i)}, y^{(i)})\right), \tag{3.25}$$

donde (3.24) se sigue de la asunción de que las observaciones son independientes.

Para el primer término de (3.25), $p(z_{i,j} = 1|y^{(i)})$ es la probabilidad a posteriori de $z^{(i)} = j$ dada la observación $y^{(i)}$:

$$\hat{z}_{i,j} = p(z_{i,j} = 1|y^{(i)}) = p(z^{(i)} = j|y^{(i)}) \tag{3.26}$$

$$\hat{z}_{i,j} = \frac{\alpha_j p(y^{(i)}|z^{(i)} = j)}{\sum_{k=1}^{K} \alpha_k p(y^{(i)}|z^{(i)} = k)} \tag{3.27}$$

Las funciones de $f(x^{(i)}, y^{(i)})$ son a lo sumo cuadráticas para $x^{(i)}$. Por lo tanto, el segundo término de (3.25) sólo depende del primer y segundo momento de los estados condicionados en $y^{(i)}$ y la componente j:

$$\hat{x}_{t|j}^{(i)} = \mathbb{E}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(x_t^{(i)}\right), \tag{3.28}$$

$$\hat{P}_{t,t|j}^{(i)} = \mathbb{E}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(P_{t,t}^{(i)}\right), \tag{3.29}$$

$$\hat{P}_{t,t-1|j}^{(i)} = \mathbb{E}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(P_{t,t-1}^{(i)}\right). \tag{3.30}$$

Estos son calculados utilizando el filtro Kalman de suavizado A.3, que estima la media y covarianza del estado $x^{(i)}$ condicionado a la observación $y^{(i)}$ y las asignaciones $z^{(i)} = j$:

$$\hat{x}_{t|j}^{(i)} = \mathbb{E}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(x_t^{(i)}\right), \tag{3.31}$$

$$\hat{V}_{t,t|j}^{(i)} = \mathrm{cov}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(x_t^{(i)}, x_t^{(i)}\right), \tag{3.32}$$

$$\hat{V}_{t,t-1|j}^{(i)} = \mathrm{cov}_{x^{(i)}|y^{(i)},z^{(i)}=j}\left(x_t^{(i)}, x_{t-1}^{(i)}\right). \tag{3.33}$$

Por lo tanto, los momentos de segundo orden de (3.29) y (3.30) son obtenidos como: $\hat{P}_{t,t|j}^{(i)} = \hat{V}_{t,t|j}^{(i)} + \hat{x}_{t|j}^{(i)}\hat{x}_{t|j}^{(i)^T}$ y $\hat{P}_{t,t-1|j}^{(i)} = \hat{V}_{t,t-1|j}^{(i)} + \hat{x}_{t|j}^{(i)}\hat{x}_{t-1|j}^{(i)^T}$.

Aplicando la esperanza de (3.13) sobre (3.15) se obtiene la función:

$$\begin{aligned}
&\mathcal{Q}(\Theta, \hat{\Theta}) = \\
&-\frac{1}{2}\sum_{j=1}^{K} \operatorname{tr}\left[R_j^{-1}\left(\Lambda_j - \Gamma_j C_j^T - C_j \Gamma_j^T + C_j \Theta_j C_j^T\right)\right] \\
&-\frac{1}{2}\sum_{j=1}^{K} \operatorname{tr}\left[S_j^{-1}\left(\eta_j - \xi_j \mu_j^T - \mu_j \xi_j^T + \hat{N}_j \mu_j \mu_j^T\right)\right] \\
&-\frac{1}{2}\sum_{j=1}^{K} \operatorname{tr}\left[Q_j^{-1}\left(\varphi_j - \Psi_j A_j^T - A_j \Psi_j^T + A_j \phi_j A_j^T\right)\right] \\
&+\sum_{j=1}^{K} \hat{N}\left[\log \alpha_j - \frac{\tau}{2}\log|R_j| - \frac{\tau-1}{2}\log|Q_j| - \frac{1}{2}\log|S_j|\right],
\end{aligned} \tag{3.34}$$

definiendo las siguientes variables por comodidad:

$$\begin{aligned}
&\hat{N}_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j}, && \Phi_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \sum_{t=1}^{\tau} \hat{P}_{t,t|j}^{(i)}, \\
&\xi_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \hat{x}_{1|j}^{(i)}, && \varphi_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \sum_{t=2}^{\tau} \hat{P}_{t,t|j}^{(i)}, \\
&\eta_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \hat{P}_{1|j}^{(i)}, && \Psi_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \sum_{t=2}^{\tau} \hat{P}_{t,t-1|j}^{(i)}, \\
&\Lambda_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \sum_{t=1}^{\tau} y_t^{(i)} {y_t}^{(i)^T}, && \phi_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \sum_{t=2}^{\tau} \hat{P}_{t-1,t-1|j}^{(i)}, \\
&\Gamma_j = \textstyle\sum_{i=1}^{N} \hat{z}_{i,j} \sum_{t=1}^{\tau} y_t^{(i)} \hat{x}_{t|j}^{(i)^T}, &&
\end{aligned} \tag{3.35}$$

con las esperanzas condicionales (3.27)-(3.30).

3.2.2. Etapa M

En la etapa M del algoritmo EM, la parametrización del modelo es obtenida maximizando la función $\mathcal{Q}$ tomando la derivada parcial con respecto a cada parámetro e igualando a cero. El problema de maximización con respecto a cada parámetro aparece de dos formas. La primera es una maximización con respecto a una matriz cuadrada X:

$$X^* = \arg\max_X -\frac{1}{2}\operatorname{tr}(X^{-1}A) - \frac{b}{2}\log|X| \tag{3.36}$$

Maximizando, tomando la derivada e igualando a cero, se obtiene la siguiente solución:

$$\frac{\partial}{\partial X} - \frac{1}{2}\operatorname{tr}(X^{-1}A) - \frac{b}{2}\log|X| = 0, \tag{3.37}$$

$$\frac{1}{2}X^{-T}A^T X^{-T} - \frac{b}{2}X^{-T} = 0, \tag{3.38}$$

$$A^T - bX^T = 0, \tag{3.39}$$

$$\Rightarrow X^* = \frac{1}{b}A. \tag{3.40}$$

La segunda forma es una maximización con respecto a una matriz X de la forma:

$$X^* = \arg\max_X -\frac{1}{2}\text{tr}\left[D(-BX^T - XB^T + XCX^T)\right], \tag{3.41}$$

donde D y C son matrices simétricas e invertibles. El máximo esta dado por:

$$\frac{\partial}{\partial X} - \frac{1}{2}\text{tr}\left[D(-BX^T - XB^T + XCX^T)\right] = 0, \tag{3.42}$$

$$-\frac{1}{2}(-DB - D^TB + D^TXC^T + DXC) = 0, \tag{3.43}$$

$$DB - DXC = 0, \tag{3.44}$$

$$\Rightarrow X^* = BC^{-1}. \tag{3.45}$$

Los parámetros óptimos se obtienen recolectando los parámetros relevantes en (3.34) y maximizando.

Por lo tanto, los parámetros de las texturas dinámicas son actualizados de acuerdo a (3.46), obteniendo el siguiente paso de actualización para cada componente j de la mixtura:

$$\begin{aligned} C_j^* &= \Gamma_j\Phi_j^{-1}, & R_j^* &= \tfrac{1}{\tau\hat{N}_j}\left(\Lambda_j - C_j^*\Gamma_j\right), \\ A_j^* &= \Psi_j\phi_j^{-1}, & Q_j^* &= \tfrac{1}{(\tau-1)\hat{N}_j}\left(\varphi_j - A_j^*\Psi_j^T\right), \\ \mu_j^* &= \tfrac{1}{\hat{N}_j}\xi_j, & S_j^* &= \tfrac{1}{\hat{N}_j}\eta_j - \mu_j^*\mu_j^{*T}, \\ \alpha_j^* &= \tfrac{\hat{N}_j}{N}. \end{aligned} \tag{3.46}$$

A continuación se presenta un resumen de EM para mixturas de texturas dinámicas.

Algorithm 1 EM para una mixtura de texturas dinámicas.

Require: N secuencias $\{y^{(i)}\}_{i=1}^N$, K componentes.
  {Inicialización}
  for $j = \{1, \ldots, K\}$ **do**
    Inicializar $\{\Theta_j, \alpha_j\}$
  end for
  repeat
    {Expectación}
    for $i = \{1, \ldots, N\}$ y $j = \{1, \ldots, K\}$ **do**
      Calcular las esperanzas de los estados ocultos con el filtro de suavizado de Kalman utilizando y_i y Θ_j
    end for
    {Maximización}
    for $j = \{1, \ldots, K\}$ **do**
      Calcular los nuevos parámetros $\{\Theta_j, \alpha_j\}$ de la componente
    end for
  until converge
Ensure: $\{\Theta_j, \alpha_j\}_{j=1}^K$

La etapa de esperanza se basa en el filtro Kalman de suavizado para calcular las esperanzas de las variables de estado oculto x_t, dadas las secuencias

observadas $y^{(i)}$, y 2) la verosimilitud de la observación $y^{(i)}$ dada la asignación de $z^{(i)}$. Luego, la etapa de maximización calcula los parámetros de máxima verosimilitud para cada componente j de textura promediando todas las secuencias $\{y^{(i)}\}_{i=1}^{N}$, ponderado por la probabilidad a posteriori de asignar $z^{(i)} = j$.

3.3. Inicialización

El algoritmo EM depende en gran medida de la inicialización para producir parámetros estimados que sean precisos. En lo que resta de la sección se presentan tres estrategias de inicialización que han demostrado empíricamente ser efectivas para el aprendizaje de mixturas de texturas dinámicas.

3.3.1. Semilla inicial

Si una agrupación inicial se encuentra disponible (por ejemplo, si se tiene un contorno inicial para segmentar regiones), cada componente de la textura es aprendido utilizando el método para la estimación de parámetros de texturas dinámicas 2.2.

3.3.2. Caminos aleatorios

Se ejecuta el algoritmo EM para diferentes inicializaciones aleatorias. Luego, se seleccionan los parámetros que mejor se ajusten a los datos teniendo en cuenta la máxima-verosimilitud. Cada inicialización se obtiene utilizando el método para la estimación de parámetros de texturas dinámicas 2.2, de muestra al azar de los datos de entrada. La desventaja de este método es que no es determinístico.

3.3.3. División de Componente

Se ejecuta el algoritmo EM incrementando la cantidad de componentes de la mixtura hasta alcanzar el número deseado de componentes:

Algorithm 2 División de Componente para EM de mixturas de texturas dinámicas.

Require: N secuencias $\{y^{(i)}\}_{i=1}^{N}$, K componentes.
  for $j = \{1, \ldots, K\}$ **do**
    if j=1 **then**
      Inicializar $\{\Theta_1\}$ utilizando la estimación de parámetros de texturas dinámicas 2.2 para una muestra de $\{y^{(i)}\}_{i=1}^{N}$.
    else
      {Selección}
      Seleccionar la componente $\{\Theta_{j'}\}$ con el mayor autovalor de Q (que peor se ajusta al espacio de estados).
      {Perturbación}
      Duplicar la componente $\{\Theta_{j'}\}$ y perturbarla escalando 1.01 la columna de C, asociada con la mayor varianza de S.
    end if
    Ejecutar EM para $\{\Theta_{j'}, \alpha_{j'}\}_{j'=1}^{j}$
  end for
Ensure: $\{\Theta_j, \alpha_j\}_{j=1}^{K}$

3.4. Clasificación

La clasificación de video puede ser una herramienta útil para descubrir patrones de alto nivel en un flujo de video, por ejemplo, eventos recurrentes, eventos con alta o baja probabilidad, eventos aislados, etc. Estas operaciones son de gran interés práctico para algunas clases de video con grupos de partículas, donde se quiere entender video adquirido de un ambiente lleno de partículas. En este contexto, la clasificación de video se puede utilizar para problemas como vigilancia, detección de cambios, sumarización de eventos o monitoreo remoto de distintos tipo de ambientes. También puede ser aplicado a una base de datos de videos para crear automáticamente una taxonomía de clases de video que puede ser utilizada para organizar la base de datos o para la obtención de video.

Utilizando mixturas de texturas dinámicas, un conjunto de secuencias de video puede ser clasificado, primero aprendiendo la mixtura que mejor se ajusta a las secuencias de video y luego asignando cada secuencia al componente de la mixtura con la mayor probabilidad a posteriori de haberla generado, esto es, etiquetando la secuencia $y^{(i)}$ con:

$$\ell_i = \arg\max_j \log p(y^{(i)}|z^{(i)} = j) + \log \alpha_j \tag{3.47}$$

3.5. Segmentación

La segmentación de videos trata el problema de descomponer una secuencia de video en una colección de regiones homogéneas. Aunque es sabido que puede ser resuelto con modelos de mixturas y el algoritmo EM, el éxito de la segmentación depende de la habilidad del modelo de mixturas para capturar la dimensión con la cual el video es estadísticamente homogéneo. Para texturas

espacio-temporales (por ejemplo, videos de humo y fuego), los modelos tradicionales de movimiento basados en mixturas no son aptos para capturar estas dimensiones, debido a su incapacidad de considerar la naturaleza estocástica del movimiento. La mixtura de texturas dinámicas extiende la aplicación de los algoritmos de segmentación basados en mixturas a videos compuestos de texturas espacio-temporales. Como en la mayoría de las propuestas anteriores basadas en mixturas aplicadas a segmentación de video, el proceso consiste de dos etapas: el modelo de la mixtura primero es aprendido y luego el video es segmentado asignando ubicaciones del video a componentes de la mixtura.

En la etapa de aprendizaje, el video es representado como un conjunto de parches. Para segmentación de texturas espacio-temporales, un parche de dimensiones $p \times p \times q$ es obtenido de cada ubicación del video (o sobre una grilla regular no solapada), donde p y q deben ser suficientemente grandes para capturar la características distintivas de las componentes. Si el borde de la segmentación no cambia en el tiempo, q puede ser igual a la longitud del video. Luego, el conjunto de parches es agrupado utilizando el algoritmo EM. La segunda etapa, la segmentación, recorre las ubicaciones del video secuencialmente. En cada posición, un parche es extraído y asignado a una componente de la mixtura, de acuerdo a (3.47). La ubicación pertenece a la región de segmentación asociada con esa clase.

Capítulo 4

Capas de texturas dinámicas

4.1. El modelo

En las capas de texturas dinámicas [CV09] se consideran videos compuestos de varias texturas, por ejemplo, la combinación de fuego, humo y agua. Este tipo de video puede ser modelado codificando cada textura como una *capa* separada, con su propia componente visual y dinámica. Diferentes regiones del volumen espacio-temporal del video son asignados a cada textura y, condicionado por esta asignación, cada región evolucionará como una textura dinámica común. O sea, el video es una composición de varias capas.

Para las capas de texturas dinámicas se utiliza el modelo gráfico que se observa en la Figura 4.1. En este modelo existen K capas, cada una de las cuales tiene una secuencia de estados ocultos $x^{(j)} = \{x_t^{(j)}\}_{t=1}^{\tau}$ que evoluciona de forma separada, donde τ es la cantidad de cuadros del video, como cualquier textura dinámica. El video $Y = \{y_i\}_{i=1}^{m}$ contiene m trayectorias de píxeles $y_i = \{y_{i,t}\}_{t=1}^{\tau}$, que son asociadas a una de las K capas a través de la variable oculta z_i. La colección de variables ocultas $Z = \{z_i\}_{i=1}^{m}$ es modelada como un campo aleatorio markoviano, para asegurar suavidad espacial al asignar las capas.

El sistema de ecuaciones de este modelo es:

$$\begin{cases} x_{t+1}^{(j)} = A^{(j)} x_t^{(j)} + v_t^{(j)} \\ y_{i,t} = C_i^{(z_i)} x_t^{(z_i)} + w_{i,t} \end{cases}$$

donde $C_i^{(j)} \in \mathbb{R}^{1\times n}$ es la matriz de transformación del estado oculto al píxel observado. Los procesos de ruido son $v_t^{(j)} \sim \mathcal{N}(0, Q^{(j)})$ y $w_{i,t} \sim \mathcal{N}(0, r^{(z_i)})$, y el estado inicial está dado por $x_1^{(j)} \sim \mathcal{N}(\xi^{(j)}, Q^{(j)})$, donde $Q^{(j)} \in S_+^n$, $r^{(j)} \in \mathbb{R}_+$, y $\xi^{(j)} \in \mathbb{R}^n$.

Si se tienen las asignaciones de las capas, el modelo de capas de texturas dinámicas es, sencillamente, una superposición de texturas dinámicas definidas

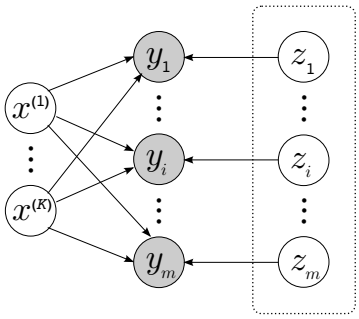

Figura 4.1: Las capas de texturas dinámicas. La variable $x^{(j)}$ es el proceso de estados ocultos de una textura dinámica e y_i es un proceso de píxeles observados. La variable z_i asigna y_j a uno de los $x^{(j)}$, y la colección $\{z_i\}$ es modelada como un campo aleatorio markoviano.

sobre diferentes regiones del volumen del video. Estimar los parámetros de las capas de texturas dinámicas se reduce a estimar los parámetros de la textura dinámica correspondiente a cada región. Cuando las asignaciones de las capas son desconocidas, los parámetros pueden ser obtenidos por medio del algoritmo EM (ver Apéndice A.2).

4.1.1. Distribución conjunta

Como es usual para modelos de mixturas, se introduce una variable indicadora $z_i^{(j)}$ con valor 1, si $z_i = j$, y 0, en caso contrario. El modelo de capas asume que los estados ocultos $X = \{x^{(j)}\}_{j=1}^{K}$ y las asignaciones de capa Z son independientes, o sea, el comportamiento de las capas es independiente de su ubicación.

Asumiendo lo anterior, los factores de la distribución conjunta son:

$$p(X, Y, Z) = p(Y|X, Z)p(X)p(Z) \tag{4.1}$$

$$= \prod_{i=1}^{m}\prod_{j=1}^{K} p(y_i|x^{(j)}, z_i = j)^{z_i^{(j)}} \prod_{j=1}^{K} p(x^{(j)})p(Z). \tag{4.2}$$

Asimismo, cada secuencia de estados es un proceso Gauss-Markov, con distribución:

$$p(x^{(j)}) = p(x_1^{(j)}) \prod_{t=2}^{\tau} p(x_t^{(j)}|x_{t-1}^{(j)}) \tag{4.3}$$

donde las densidades de los estados son:

$$p(x_1^{(j)}) = G(x_1^{(j)}, \xi^{(j)}, Q^{(j)}) \tag{4.4}$$

$$p(x_t^{(j)}|x_{t-1}^{(j)}) = G(x_t^{(j)}, A^{(j)}x_{t-1}^{(j)}, Q^{(j)}) \tag{4.5}$$

y $G(x, \mu, \Sigma) = (2\pi)^{-n/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}\|x-\mu\|_\Sigma^2}$ es una distribución gaussiana n-dimensional de media μ y covarianza Σ, y $\|x\|_\Sigma^2 = x^T\Sigma^{-1}x$ es la distancia

Mahalanobis con respecto a Σ.

Las distribuciones de las trayectorias de los píxeles, condicionadas a una secuencia de estados y a una asignación de capas, son independientes y tienen distribución:

$$p(y_i|x^{(j)}, z_i = j) = \prod_{t=1}^{\tau} p(y_{i,t}|x_t^{(j)}, z_i = j) \tag{4.6}$$

$$p(y_{i,t}|x_t^{(j)}, z_i = j) = G(y_{i,t}, C_i^{(j)} x_t^{(j)}, r^{(j)}) \tag{4.7}$$

Finalmente, las asignaciones de capa están distribuidas como:

$$p(Z) = \frac{1}{\mathcal{Z}_Z} \prod_{i=1}^{m} V_i(z_i) \prod_{(i,i')\in\varepsilon} V_{i,i'}(z_i, z_{i'}), \tag{4.8}$$

donde ε es el conjunto de aristas del campo aleatorio markoviano, $\mathcal{Z}_Z$ una constante de normalización (función de partición), y V_i y $V_{i,i'}$ son las funciones de potencia de la forma:

$$V_i(z_i) = \prod_{j=1}^{K} (\alpha_i^{(j)})^{z_i^{(j)}} = \begin{cases} \alpha_i^{(1)}, z_i = 1, \\ \vdots \\ \alpha_i^{(K)}, z_i = K, \end{cases} \tag{4.9}$$

$$V_{i,i'}(z_i, z_{i'}) = \gamma_2 \prod_{j=1}^{K} \left(\frac{\gamma_1}{\gamma_2}\right)^{z_i^{(j)} z_{i'}^{(j)}} = \begin{cases} \gamma_1, z_i = z_{i'}, \\ \gamma_2, z_i \neq z_{i'}, \end{cases} \tag{4.10}$$

donde V_i es la probabilidad a priori de cada capa, mientras que $V_{i,i'}$ da mayor probabilidad a configuraciones de vecindarios donde los píxeles están en la misma capa.

En este trabajo se trata al campo aleatorio markoviano como un a priori sobre Z, que controla la suavidad de las capas. Los parámetros de las funciones de potencia de cada capa pueden ser aprendidos, pero esto no es necesario.

4.2. Estimación de los parámetros con EM

Dado un video de entrenamiento Y, los parámetros $\Theta = \{C_i^{(j)}, A^{(j)}, r^{(j)}, Q^{(j)}\}_{j=1}^{K}$ de las capas de texturas dinámicas son aprendidos utilizando la verosimilitud máxima:

$$\Theta^* = \arg\max_{\Theta} \log p(Y) \tag{4.11}$$

$$= \arg\max_{\Theta} \log \sum_{X,Z} p(Y, X, Z) \tag{4.12}$$

Ya que la verosimilitud de los datos depende de variables ocultas (la secuencia de estados X y las asignaciones de capa Z), este problema se puede resolver por medio del algoritmo EM A.2, que itera entre:

$$\text{Etapa E:} \quad \mathcal{Q}(\Theta; \hat{\Theta}) = \mathbb{E}_{X,Y|Y,\hat{\Theta}}(\log p(X, Y, Z; \Theta)) \tag{4.13}$$

$$\text{Etapa M:} \quad \hat{\Theta}^* = \arg\max_{\Theta} \mathcal{Q}(\Theta; \hat{\Theta}) \tag{4.14}$$

donde $\log p(X, Y, Z; \Theta)$ es el logaritmo de la verosimilitud completa de los datos parametrizado por Θ, y $\mathbb{E}_{X,Y|Y;\hat{\Theta}}$ es la esperanza de X y Z, condicionada en Y y parametrizada con la estimación actual de $\hat{\Theta}$.

4.2.1. Logaritmo de la verosimilitud completa

Partiendo de (4.2), el logaritmo de la verosimilitud completa de los datos es:

$$\begin{aligned}\log p(X, Y, Z) = &\sum_{i=1}^{m}\sum_{j=1}^{K} z_i^{(j)} \sum_{t=1}^{\tau} \log p(y_{i.t}|x_t^{(j)}, z_i = j) \\ &+ \sum_{j=1}^{K}\left(\log p(x_1^{(j)}) + \sum_{t=2}^{\tau} \log p(x_t^{(j)}|x_{t-1}^{(j)})\right) \\ &+ \log p(Z)\end{aligned} \tag{4.15}$$

Usando los cálculos (4.4), (4.5) y (4.7), y eliminando los términos que no dependen de los parámetros de Θ (que no se utilizan en el etapa M) se obtiene:

$$\begin{aligned}\log p(X, Y, Z) = &-\frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{m} z_i^{(j)} \sum_{t=1}^{\tau}\Bigg(\|y_{i,t} - C_i^{(j)} x_t^{(j)}\|^2_{r^{(j)}} \\ &+ \log r^{(j)}\Bigg) + \frac{1}{2}\sum_{j=1}^{K}\Bigg(\|x_1^{(j)} - \xi^{(j)}\|^2_{Q^{(j)}} \\ &+ \sum_{t=2}^{\tau}\|x_t^{(j)} - A^{(j)} x_{t-1}^{(j)}\|^2_{Q^{(j)}} + \tau \log|Q^{(j)}|\Bigg),\end{aligned} \tag{4.16}$$

donde $\|x\|^2_\Sigma = x^T \Sigma^{-1} x$. Se puede ver que $p(Z)$ puede ser ignorado, ya que los parámetros del campo aleatorio markoviano son constantes. Finalmente, el logaritmo de la probabilidad completa es:

$$\begin{aligned}\log p(X, Y, Z) = &-\frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{m} z_i^{(j)} \sum_{t=1}^{\tau}\frac{1}{r^{(j)}}\Bigg({y_{i,t}}^2 - 2y_{i,t} C_i^{(j)} x_t^{(j)} \\ &+ C_i^{(j)} P_{t,t}^{(j)} {C_i^{(j)}}^T\Bigg) - \frac{1}{2}\sum_{j=1}^{K}\mathrm{tr}\Bigg[{Q^{(j)}}^{-1}\Bigg(P_{1,1}^{(j)} \\ &- x_1^{(j)}{\xi^{(j)}}^T - \xi^{(j)}{x_1^{(j)}}^T + \xi^{(j)}{\xi^{(j)}}^T\Bigg)\Bigg] \\ &- \frac{1}{2}\sum_{j=1}^{K}\sum_{t=2}^{\tau}\mathrm{tr}\Bigg[{Q^{(j)}}^{-1}\Bigg(P_{t,t}^{(j)} + P_{t,t-1}^{(j)}{A^{(j)}}^T \\ &- A^{(j)}{P_{t,t-1}^{(j)}}^T + A^{(j)} P_{t-1,t-1}^{(j)}{A^{(j)}}^T\Bigg)\Bigg] \\ &- \frac{\tau}{2}\sum_{j=1}^{K}\sum_{i=1}^{m} z_i^{(j)}\log r^{(j)} - \frac{\tau}{2}\sum_{j=1}^{K}\log|Q^{(j)}|\end{aligned} \tag{4.17}$$

donde $P_{t,t}^{(j)} = x_t^{(j)}{x_t^{(j)}}^T$ y $P_{t,t-1}^{(j)} = x_t^{(j)}{x_{t-1}^{(j)}}^T$

4.2.2. Etapa E

Si se examina (4.17), se puede observar que la etapa E del algoritmo (4.13) requiere esperanzas condicionales de dos formas:

$$\begin{aligned}\mathbb{E}_{X,Z|Y}[f(x^{(j)})] &= \mathbb{E}_{X|Y}[f(x^{(j)})], \\ \mathbb{E}_{X,Z|Y}[z_i^{(j)} f(x^{(j)})] &= \mathbb{E}_{Z|Y}[z_i^{(j)}]\mathbb{E}_{X|Y,z_i=j}[f(x^{(j)})],\end{aligned} \tag{4.18}$$

para alguna función f de $x^{(j)}$, donde $\mathbb{E}_{X|Y,z_i=j}$ es la esperanza condicional de X dada la observación Y, y que el iésimo píxel pertenezca a la capa j. O sea, la etapa E requiere:

$$\begin{array}{rclcrcl} \hat{x}_t^{(j)} & = & \mathbb{E}_{X|Y}[x_t^{(j)}], & & \hat{P}_{t,t}^{(j)} & = & \mathbb{E}_{X|Y}[P_{t,t}^{(j)}], \\ \hat{z}_i^{(j)} & = & \mathbb{E}_{Z|Y}[z_i^{(j)}], & & \hat{P}_{t,t-1}^{(j)} & = & \mathbb{E}_{X|Y}[P_{t,t-1}^{(j)}], \\ \hat{x}_{t|i}^{(j)} & = & \mathbb{E}_{X|Y,z_i=j}[x_t^{(j)}], & & \hat{P}_{t,t|i}^{(j)} & = & \mathbb{E}_{X|Y,z_i=j}[P_{t,t}^{(j)}]. \end{array} \tag{4.19}$$

Definiendo las siguientes variables por comodidad:

$$\begin{array}{rclcrcl} \phi_1^{(j)} & = & \sum_{t=1}^{\tau-1} \hat{P}_{t,t}^{(j)}, & & \phi_2^{(j)} & = & \sum_{t=2}^{\tau} \hat{P}_{t,t}^{(j)}, \\ \psi^{(j)} & = & \sum_{t=2}^{\tau} \hat{P}_{t,t-1}^{(j)}, & & \hat{N}_j & = & \sum_{i=1}^{m} \hat{z}_i^{(j)}, \\ \Phi_i^{(j)} & = & \sum_{t=1}^{\tau} \hat{P}_{t,t|i}^{(j)}, & & \gamma^{(j)} & = & \sum_{t=1}^{\tau} \hat{x}_{t|i}^{(j)}, \\ \Gamma_i^{(j)} & = & \sum_{t=1}^{\tau} y_{i,t} \hat{x}_{t|i}^{(j)}, & & & & \end{array} \tag{4.20}$$

y sustituyendo (4.20) y (4.17) en (4.13), se obtiene como resultado:

$$\begin{aligned}\mathcal{Q}(\Theta;\hat{\Theta}) = &-\frac{1}{2}\sum_{j=1}^{K}\frac{1}{r^{(j)}}\sum_{i=1}^{m}\hat{z}_i^{(j)}\Bigg(\sum_{t=1}^{\tau} {y_{i,t}}^2 \\ &- 2C_i^{(j)}\Gamma_i^{(j)} + C_i^{(j)}\Phi_i^{(j)}{C_i^{(j)}}^T\Bigg) - \frac{1}{2}\sum_{j=1}^{K}\mathrm{tr}\Bigg[{Q^{(j)}}^{-1} \\ &\Bigg(P_{1,1}^{(j)} - \hat{x}_1^{(j)}{\xi^{(j)}}^T - \xi^{(j)}{\hat{x}_1^{(j)}}^T + \xi^{(j)}{\xi^{(j)}}^T + \phi_2^{(j)} \\ &- \psi^{(j)}{A^{(j)}}^T - A^{(j)}{\psi^{(j)}}^T + A^{(j)}\phi_1^{(j)}{A^{(j)}}^T\Bigg)\Bigg] \\ &- \frac{\tau}{2}\sum_{j=1}^{K}\hat{N}_j \log r^{(j)} - \frac{\tau}{2}\sum_{j=1}^{K}\log|Q^{(j)}|\end{aligned} \tag{4.21}$$

Dado que no se sabe a qué capa es asignado cada píxel, el cálculo de las esperanzas (4.19) requiere la marginalización sobre todas las configuraciones de Z. Esto significa que la función $\mathcal{Q}$ es intratable. Dos posibles aproximaciones se discuten en las secciones 4.3 y 4.4.

4.2.3. Etapa M

La maximización de la función $\mathcal{Q}$ con respecto a los parámetros del modelo origina dos tipos de problemas de optimización. El primero es una maximización con respecto a una matriz cuadrada X de la forma:

$$X^* = \arg\max_X -\frac{1}{2}\mathrm{tr}(X^{-1}A) - \frac{b}{2}\log|X| \tag{4.22}$$

Tomando las derivadas e igualando a cero se obtiene:

$$\frac{\partial}{\partial X} - \frac{1}{2}\mathrm{tr}(X^{-1}A) - \frac{b}{2}\log|X| = 0, \tag{4.23}$$

$$\frac{1}{2}X^{-T}A^{T}X^{-T} - \frac{b}{2}X^{-T} = 0, \tag{4.24}$$

$$A^{T} - bX^{T} = 0, \tag{4.25}$$

$$\Rightarrow X^{*} = \frac{1}{b}A. \tag{4.26}$$

El segundo es una maximización con respecto a una matriz X de la forma:

$$X^{*} = \arg\max_{X} -\frac{1}{2}\mathrm{tr}\left[D(-BX^{T} - XB^{T} + XCX^{T})\right], \tag{4.27}$$

donde D y C son matrices simétricas e invertibles. El máximo esta dado por:

$$\frac{\partial}{\partial X} - \frac{1}{2}\mathrm{tr}\left[D(-BX^{T} - XB^{T} + XCX^{T})\right] = 0, \tag{4.28}$$

$$-\frac{1}{2}(-DB - D^{T}B + D^{T}XC^{T} + DXC) = 0, \tag{4.29}$$

$$DB - DXC = 0, \tag{4.30}$$

$$\Rightarrow X^{*} = BC^{-1}. \tag{4.31}$$

Los parámetros óptimos se obtienen recolectando los parámetros relevantes en (4.21) y maximizando.

Por lo tanto, la etapa de M (4.14) actualiza las estimaciones de los parámetros maximizando la función $\mathcal{Q}$. Como es usual, un máximo (local) se encuentra tomando la derivada parcial con respecto a cada parámetro e igualando a cero, lo que nos lleva a:

$$\begin{aligned}
C_i^{(j)*} &= {\Gamma_i^{(j)}}^{T}{\Phi_i^{(j)}}^{-1}, \\
A^{(j)*} &= \psi^{(j)}{\phi_1^{(j)}}^{-1}, \\
\xi^{(j)*} &= \hat{x}_1^{(j)}, \\
r^{(j)*} &= \frac{1}{\tau\hat{N}_j}\sum_{i=1}^{m} z_i^{(j)}\left(\sum_{t=1}^{\tau} {y_{i,t}}^{2} - C_i^{(j)*}\Gamma_i^{(j)}\right), \\
Q^{(j)*} &= \frac{1}{\tau}\left(P_{1,1}^{(j)} - \xi^{(j)*}\left(\xi^{(j)*}\right)^{T} + \phi_2^{(j)} - A^{(j)*}{\psi^{(j)}}^{T}\right),
\end{aligned} \tag{4.32}$$

4.3. Inferencia aproximada por muestreo de Gibbs

Las esperanzas (4.19) necesitan probabilidades condicionales intratables. Por ejemplo, $P(X|Y) = \sum_Z P(X, Z|Y)$ requiere todas las posibles configuraciones de Z, una operación de complejidad exponencial con las dimensiones de un campo aleatorio markoviano, intratable aún para tamaños de ventana pequeñas. Una solución común a este problema es utilizar un generador de muestras de Gibbs [GG87] para obtener muestras de una distribución a posteriori de $p(X, Z|Y)$ y aproximar las esperanzas promediando las muestras obtenidas. Dado un estado inicial $\tilde{Z}$, el muestreo de Gibbs en cada iteración alterna entre muestrear $\tilde{X}$ de $p(X|\tilde{Z}, Y)$ y muestrear $\tilde{Z}$ de $p(Z|\tilde{X}, Y)$.

4.3.1. Muestrear de $p(Z|X,Y)$

Usando las regla de Bayes la distribución condicional $p(Z|X,Y)$ puede ser reescrita como:

$$\begin{aligned} p(Z|X,Y) &= \frac{p(X,Y,Z)}{p(X,Y)} = \frac{p(Y|X,Z)p(X)p(Z)}{p(X,Y)} \\ &\propto p(Y|X,Z)p(Z) \propto \prod_{i=1}^{m}\prod_{j=1}^{K} p(y_i|x^{(j)}, z_i = j)^{z_i^{(j)}} \\ &\quad \cdot \left[\prod_{i=1}^{m} V_i(z_i) \prod_{(i,i')\in\varepsilon} V_{i,i'}(z_i, z_{i'})\right] \\ &= \prod_{i=1}^{m}\prod_{j=1}^{K} \left[\alpha_i^{(j)} p(y_i|x^{(j)}, z_i = j)\right]^{z_i^{(j)}} \prod_{(i,i')\in\varepsilon} V_{i,i'}(z_i, z_{i'}). \end{aligned} \tag{4.33}$$

Entonces, $p(Z|X,Y)$ es equivalente a la función de probabilidad del campo aleatorio markoviano de (4.8), pero con la función de potencia modificada $\tilde{V}_i(z_i) = \alpha_i^{(j)} p(y_i|x^{(j)}, z_i = j)$. Por lo tanto, las muestras de $p(Z|X,Y)$ pueden ser obtenidas utilizando Monte Carlo para una grilla de un campo aleatorio markoviano [GG87].

Gibbs MCMC

$$\pi(w) = \frac{e^{-\beta\xi(w)}}{\sum_w e^{-\beta\xi(w)}} \tag{4.34}$$

$$= \frac{e^{-\beta\xi(w)}}{Z} \tag{4.35}$$

donde $w \in \Omega$ son todas las configuraciones posibles del sistema, $\xi(w) = \sum_C V_C(w)$ es la energía potencial de w siendo V_C la función de potencia, y $\beta = 1/KT$ siendo K la constante de Boltzmann y T la temperatura absoluta. La temperatura se utiliza para elegir muestras que no son necesariamente mejores que la muestra anterior.
Para aproximar Y, una variable de interés, tendríamos que calcular:

$$\tilde{Y} = \int_\Omega Y(w) d\pi(w) = \frac{\sum_w Y(w) e^{-\beta\xi(w)}}{\sum_w e^{-\beta\xi(w)}} \tag{4.36}$$

Ya que esto no puede ser realizado de forma analítica utilizamos el muestreo para obtener un valor aproximado de Y. Para ello obtenemos muestras $w_1, w_2, \ldots, w_R$ de π y $\tilde{Y}$ es aproximado utilizando medias ergodics:

$$\tilde{Y} \approx \frac{1}{R}\sum_{r=1}^{R} Y(w_r) \tag{4.37}$$

La toma de muestras es realizada de la siguiente forma. Dado un estado del sistema al "instante" t, $X(t)$, se toma otra configuración η y se calcula el cambio

de energía $\Delta\xi = \xi(\eta) - \xi(X(t))$ y la cantidad:

$$q = \frac{\pi(\eta)}{\xi(X(t))} = e^{\Delta\xi} \tag{4.38}$$

Si $q > 1$ se toma η, por lo que $X(t+1) = \eta$, mientras que si $q \leq 1$ la transición es hecha con probabilidad q. O sea, se elige $0 \leq \delta \leq 1$ de forma uniforme y se elige $X(t+1) = \eta$ si $\delta \leq q$ y $X(t+1) = X(t)$ en caso contrario.
Generalmente:

$$T(k) = \frac{C}{\log(1+k)} \tag{4.39}$$

con $k \in \{1 \dots K\}$ y C es una constante que comúnmente está en $\{3, 4\}$.

$$V_C = \begin{cases} \frac{1}{3}, & f_s = f_r \text{ (están en la misma capa)} \\ -\frac{1}{3}, & f_s \neq f_r \text{ (están en distintas capas)} \end{cases} \tag{4.40}$$

Se toma una posicion de la grilla y se obtiene una muestra al azar para esa posición y luego se calcula la probabilidad para el cambio al nuevo estado con la nueva muestra.

4.3.2. Muestrear de $p(X|Z,Y)$

Dados las asignaciones de capas Z, los píxeles son determinísticamente asignados a procesos de estados. Por conveniencia, se define $\mathcal{I}_j = \{i|z_i = j\}$ como el conjunto de índices asignados a la capa j, y $Y_j = \{y_i|i \in \mathcal{I}_j\}$ como el conjunto correspondiente a los valores de los píxeles. Condicionando en Z, obtenemos:

$$p(X, Y|Z) = \prod_{j=1}^{K} p(x^{(j)}, Y_j|Z). \tag{4.41}$$

Notar que $p(x^{(j)}, Y_j|Z)$ es la distribución de un sistema dinámico lineal con parámetros $\tilde{\Theta}_j = \{A^{(j)}, Q^{(j)}, \tilde{C}^{(j)}, r^{(j)}, \xi^{(j)}\}$, donde $\tilde{C}^{(j)} = \left[C_i^{(j)}\right]_{i \in \mathcal{I}_j}$. Si marginalizamos (4.41) con respecto a X obtenemos:

$$p(Y|Z) = \prod_{j=1}^{K} p(Y_j|Z), \tag{4.42}$$

donde $p(Y_j|Z)$ es la probabilidad de observar Y_j del sistema dinámico lineal $\tilde{\Theta}_j$. Finalmente, usando la regla de Bayes:

$$p(X|Y, Z) = \frac{p(X, Y|Z)}{p(Y|Z)} = \frac{\prod_{j=1}^{K} p(x^{(j)}, Y_j|Z)}{\prod_{j=1}^{K} p(Y_j|Z)} \tag{4.43}$$

$$= \prod_{j=1}^{K} p(x^{(j)}|Y_j, Z) \tag{4.44}$$

O sea, tomar muestras de $p(X|Y,Z)$ se reduce a tomar muestras de una secuencia de estados $x^{(j)}$ para cada $p(x^{(j)}|Y_j, Z)$, que es la distribución condicional de $x^{(j)}$ dados los píxeles de Y_j parametrizados por $\tilde{\Theta}_j$. Un algoritmo para calcular eficientemente estas secuencias es el filtro de Kalman.

4.3.3. Inferencia aproximada

El muestreo de Gibbs primero es "inicializado" con 100 iteraciones. Esto permite a las muestras para $\{\tilde{X}, \tilde{Z}\}$ converger a una distribución a posteriori $p(X, Z|Y)$ verdadera. Las muestras subsecuentes, obtenidas luego cada cinco iteraciones, son usadas para aproximar la inferencia.

Esperanzas aproximadas

Las esperanzas (4.19) son aproximadas promediando las muestras obtenidas del muestreo de Gibbs, e.j., $\mathbb{E}_{X|Y}[x_t^{(j)}] \approx \frac{1}{S}\sum_{s=1}^{S}[\hat{x}_t^{(j)}]_s$, donde $[\hat{x}_t^{(j)}]_s$ es el valor de $x_t^{(j)}$ en la s-ésima muestra, y S es el número de muestras.

Cota inferior en $p(Y)$

La convergencia del algoritmo EM es monitoreada siguiendo la probabilidad de $p(Y)$ de los datos observados. Mientras que esta probabilidad es intratable, una cota inferior puede ser calculada sumando todas las configuraciones de $\tilde{Z}$ visitadas por el muestreo de Gibbs:

$$p(Y) = \sum_{Z} p(Y|Z)p(Z) \geq \sum_{\tilde{Z}\in\mathcal{Z}_G} p(Y|\tilde{Z})p(\tilde{Z}), \tag{4.45}$$

donde $\mathcal{Z}_G$ es el conjunto de estados únicos de $\tilde{Z}$ visitados por el muestreo de Gibbs, $p(Z)$ es (4.8) y $p(Y|\tilde{Z})$ es (4.42), donde para cada observación Y_j, la probabilidad de $p(Y_j|Z)$ es obtenida usando el filtro de Kalman con parámetros Θ_j A.3. Como $\mathcal{Z}_G$ tiende a las configuraciones con mayor probabilidad, la cota en (4.45) es una buena aproximación para monitorear la convergencia.

Asignación de capa

Finalmente, la segmentación requiere la solución MAP $\{X^*, Z^*\} = \arg\max_{X,Z} p(X, Z|Y)$. Esto es calculado con "annealing" deterministico [GG87].

4.4. Inferencia por aproximación variacional

Consiste en aproximar la distribución a posteriori $p(X, Z|Y)$ por una aproximación $q(X, Z)$ dentro de una clase $\mathcal{F}$ de distribuciones tratables. Dada una observación Y, la aproximación variacional óptima minimiza la divergencia de Kullback-Leibler (KL), entre las dos probabilidades a posteriori A.4:

$$q^*(X, Z) = \arg\min_{q\in\mathcal{F}} \mathrm{KL}(q(X, Z)\|p(X, Z|Y)) \tag{4.46}$$

Ya que el logaritmo de la probabilidad de $p(Y)$ es constante para una observación Y:

$$\mathrm{KL}(q(X, Y)\|p(X, Z|Y)) = \int q(X, Z)\log\frac{q(X, Z)}{p(X, Z|Y)}dXdZ \tag{4.47}$$

$$= \int q(X, Z)\log\frac{q(X, Z)p(Y)}{p(X, Z, Y)}dXdZ \tag{4.48}$$

$$= \mathcal{L}(q(X, Z)) + \log(p(Y)) \tag{4.49}$$

donde:

$$\mathcal{L}(q(X,Z)) = \int q(X,Z)\log\frac{q(X,Z)}{p(X,Z,Y)}dXdZ \tag{4.50}$$

$$= \mathrm{KL}(q(X,Y)\|p(X,Z,Y)) \tag{4.51}$$

Por lo tanto, el problema de optimización de (4.46) es equivalente a:

$$q^*(X,Z) = \arg\min_{q\in\mathcal{F}} \mathcal{L}(q(X,Z)) \tag{4.52}$$

4.4.1. Aproximación de la distribución a posteriori

Ya que se debe marginalizar sobre Z, la distribución a posteriori exacta es intratable. Una aproximación tratable puede ser obtenida asumiendo independencia entre las asignaciones a píxeles z_i y las variables de estado $x^{(j)}$:

$$q(X,Z) = \prod_{j=1}^{K} q(x^{(j)}) \prod_{i=1}^{m} q(z_i). \tag{4.53}$$

Si sustituimos en (4.50) obtenemos:

$$\mathcal{L}(q(X,Z)) = \int \prod_{j=1}^{K} q(x^{(j)}) \prod_{i=1}^{m} q(z_i) \log\frac{\prod_{j=1}^{K} q(x^{(j)}) \prod_{i=1}^{m} q(z_i)}{p(X,Y,Z)}dXdZ. \tag{4.54}$$

La ecuación (4.54) es minimizada optimizando secuencialmente cada uno de los factores $q(x^{(j)})$ y $q(z_i)$, mientras se mantiene a los otros constantes. Por conveniencia se define la variable $W = \{X, Z\}$. Reescribiendo (4.54) en función de un solo factor $q(w_l)$, mientras mantenemos a los otros constantes:

$$\mathcal{L}(q(W)) = \int q(w_l) \prod_{k\neq l} q(w_k) \log\frac{q(w_l)\prod_{k\neq l} q(w_k)}{p(W,Y)}dW \tag{4.55}$$

$$= \int q(w_l) \prod_{k\neq l} q(w_k) \log q(w_l) dW \tag{4.56}$$

$$+ \int q(w_l) \prod_{k\neq l} q(w_k) \log \prod_{k\neq l} q(w_k) dW$$

$$- \int q(w_l) \prod_{k\neq l} q(w_k) \log p(W,Y) dW$$

$$\propto \int q(w_l)\log q(w_l)dw_l - \int q(w_l) \int \prod_{k\neq l} q(w_k) \log p(W,Y) dW \tag{4.57}$$

$$= \int q(w_l)\log q(w_l)dw_l - \int q(w_l)\log \hat{p}(w_l,Y)dw_l \tag{4.58}$$

$$= \mathrm{KL}(q(w_l)||\hat{p}(w_l,Y)) \tag{4.59}$$

donde en (4.57) se eliminan los términos que no dependen de $q(w_l)$ (dado que no ejercen ningún efecto en la optimización):

$$\int q(w_l) \prod_{k\neq l} q(w_k) \log \prod_{k\neq l} q(w_k) dW = \int \prod_{k\neq l} q(w_k) \log \prod_{k\neq l} q(w_k) dW_{k\neq l}, \tag{4.60}$$

y se define $\hat{p}(w_l, Z)$ como:

$$\log \hat{p}(w_l, Z) \propto \mathbb{E}_{W_{k\neq l}}[\log p(W, Y)] \tag{4.61}$$

donde:

$$\mathbb{E}_{W_{k\neq l}}[\log p(W, Y)] = \int \prod_{k\neq l} q(w_k) \log p(W, Y) dW_{k\neq l} \tag{4.62}$$

Es fácil ver que (4.58) se minimiza cuando $q^*(w_l) = \hat{p}(w_l, Y)$. A continuación se derivan los factores $q(x^{(j)})$ y $q(z_i)$ en particular. Hasta que las formas de los factores no sean conocidas, se ignoran las constantes de normalización para facilitar los cálculos.

Optimización de $q(x^{(j)})$

Reescribiendo (4.61) con $w_l = x^{(j)}$:

$$\log q^*(x^{(j)}) = \log \hat{p}(x^{(j)}, Y) \propto \mathbb{E}_{Z, X_{k\neq j}}[\log p(X, Y, Z)] \tag{4.63}$$

$$\propto \mathbb{E}_{Z, X_{k\neq l}} \left[\sum_{i=1}^{m} z_i^{(j)} \log p(y_i | x^{(j)}, z_i = j) + \log(x^{(j)}) \right] \tag{4.64}$$

$$= \sum_{i=1}^{m} \mathbb{E}_{z_i}[z_i^{(j)}] \log p(y_i | x^{(j)}, z_i = j) + \log(x^{(j)}), \tag{4.65}$$

donde los términos de que no dependen de $x^{(j)}$ han sido eliminados.

Ahora, definiendo $h_i^{(j)} = \mathbb{E}_{z_i}[z_i^{(j)}] = \int q(z_i) z_i^{(j)} dz_i$, el término de normalización esta dado por:

$$\mathcal{Z}_q^{(j)} = \int p(x^{(j)}) \prod_{i=1}^{m} p(y_i | x^{(j)}, z_i = j)^{h_i^{(j)}} dx^{(j)} \tag{4.66}$$

y tomando log:

$$\log \mathcal{Z}_q^{(j)} = \log \int p(x^{(j)}) \prod_{i=1}^{m} \prod_{t=1}^{\tau} p(y_{i,t} | x_t^{(j)}, z_i = j)^{h_i^{(j)}} dx^{(j)} \tag{4.67}$$

Se define $\mathcal{I}_j = \{i | h_i^{(j)} > 0\}$ ya que el término $p(y_{i,t} | x_t^{(j)}, z_i = j)^{h_i^{(j)}}$ no afecta la integral cuando $h_i^{(j)} = 0$. Aplicando se obtiene:

$$\log \mathcal{Z}_q^{(j)} = \log \int p(x^{(j)}) \prod_{i\in\mathcal{I}_j} \prod_{t=1}^{\tau} p(y_{i,t} | x_t^{(j)}, z_i = j)^{h_i^{(j)}} dx^{(j)} \tag{4.68}$$

donde:

$$p(y_{i,t} | x_t^{(j)}, z_i = j)^{h_i^{(j)}} = G(y_{i,t}, C_i^{(j)} x_t^{(j)}, r^{(j)})^{h_i^{(j)}} \tag{4.69}$$

$$= (2\pi r^{(j)})^{-\frac{1}{2} h_i^{(j)}} \left(\frac{2\pi r^{(j)}}{h_i^{(j)}} \right)^{\frac{1}{2}} \tag{4.70}$$

$$\cdot G\left(y_{i,t}, C_i^{(j)} x_t^{(j)}, \frac{r^{(j)}}{h_i^{(j)}} \right).$$

Por conveniencia, se define el sistema dinámico lineal sobre $\mathcal{I}_j$ parametrizado por $\hat{\Theta}_j = \{A^{(j)}, Q^{(j)}, \hat{C}^{(j)}, \hat{R}^{(j)}, \xi^{(j)}\}$, donde $\hat{C}^{(j)} = [C_i^{(j)}]_{i\in\mathcal{I}_j}$ y $\hat{R}^{(j)}$ es una matriz diagonal con valores $\hat{r}_i^{(j)} = \frac{r^{(j)}}{h_i^{(j)}}$ para cada $i \in \mathcal{I}_j$. Este sistema dinámico lineal tiene verosimilitud condicional para la observación:

$$\hat{p}(y_{i,t}|x_t^{(j)}, z_i = j) = G(y_{i,t}, C_i^{(j)}, x_t^{(j)}, \hat{r}_i^{(j)}), \tag{4.71}$$

entonces se puede reescribir:

$$\begin{aligned} p(y_{i,t}|x_t^{(j)}, z_i = j)^{h_i^{(j)}} &= (2\pi r^{(j)})^{\frac{1}{2}(1-h_i^{(j)})}(h_i^{(j)})^{-\frac{1}{2}} \\ &\quad \cdot \hat{p}(y_{i,t}|x_t^{(j)}, z_i = j) \end{aligned} \tag{4.72}$$

y también:

$$\begin{aligned} \log \mathcal{Z}_q^{(j)} = \log \int p(x^{(j)}) \prod_{i\in\mathcal{I}_j} \prod_{t=1}^{\tau} \Big[&(2\pi r^{(j)})^{\frac{1}{2}(1-h_i^{(j)})}(h_i^{(j)})^{-\frac{1}{2}} \\ &\cdot \hat{p}(y_{i,t}|x_t^{(j)}, z_i = j)\Big] dx^{(j)}. \end{aligned} \tag{4.73}$$

Bajo el sistema dinámico lineal $\hat{\Theta}_j$, la verosimilitud de $Y_j = [y_i]_{i\in\mathcal{I}_j}$ es:

$$\hat{p_j}(Y_j) = \int p(x^{(j)}) \prod_{i\in\mathcal{I}_j} \prod_{t=1}^{\tau} \hat{p}(y_{i,t}|x_t^{(j)}, z_i = j) dx^{(j)}, \tag{4.74}$$

y por lo tanto:

$$\log \mathcal{Z}_q^{(j)} = \frac{\tau}{2} \sum_{i\in\mathcal{I}_j} (1 - h_i^{(j)}) \log(2\pi r^{(j)}) - \frac{\tau}{2} \sum_{i\in\mathcal{I}_j} \log h_i^{(j)} + \log \hat{p}_j(Y_j). \tag{4.75}$$

Optimización de $q(z_i)$

Reescribiendo (4.61) con $w_l = z_i$ y eliminando los términos que no dependen de z_i tenemos que:

$$\log q^*(z_i) = \log \hat{p}(z_i, Y) \propto \mathbb{E}_{X,Z_{k\neq i}}[\log p(X, Y, Z)] \tag{4.76}$$

$$\propto \mathbb{E}_{X,Z_{k\neq i}}\Big[\sum_{i=1}^{m} z_i^{(j)} \log p(y_i|x^{(j)}, z_i = j) \tag{4.77}$$

$$+ \log\big(V_i(z_i) \prod_{(i,i')\in\mathcal{E}} V_{i,i'}(z_i, z_{i'})\big)\Big]$$

$$= \sum_{j=1}^{K} z_i^{(j)} \mathbb{E}_{x^{(j)}}[\log p(y_i|x^{(j)}, z_i = j)] \tag{4.78}$$

$$+ \sum_{(i,i')\in\mathcal{E}} \mathbb{E}_{z_{i'}}[\log V_{i,i'}(z_i, z_{i'})] + \log V_i(z_i).$$

Si miramos los últimos dos términos, tenemos:

$$\sum_{(i,i')\in\mathcal{E}} \mathbb{E}_{z_{i'}}[\log V_{i,i'}(z_i, z_{i'})] + \log V_i(z_i) \tag{4.79}$$

$$= \sum_{(i,i')\in\mathcal{E}} \mathbb{E}_{z_{i'}} \left[\sum_{j=1}^{K} z_i^{(j)} z_{i'}^{(j)} \log \frac{\gamma_1}{\gamma_2} + \log \gamma_2 \right] + \sum_{j=1}^{K} z_i^{(j)} \log \alpha_i^{(j)} \tag{4.80}$$

$$= \sum_{j=1}^{K} z_i^{(j)} \left(\sum_{(i,i')\in\mathcal{E}} \mathbb{E}_{z_{i'}}[z_{i'}^{(j)}] \log \frac{\gamma_1}{\gamma_2} + \log \alpha_i^{(j)} \right), \tag{4.81}$$

por lo cual, finalmente obtenemos:

$$\log q^*(z_i) = \sum_{j=1}^{K} z_i^{(j)} \mathbb{E}_{x^{(j)}}[\log p(y_i|x^{(j)}, z_i = j)] \tag{4.82}$$

$$+ \sum_{j=1}^{K} z_i^{(j)} \left(\sum_{(i,i')\in\mathcal{E}} \mathbb{E}_{z_{i'}}[z_{i'}^{(j)}] \log \frac{\gamma_1}{\gamma_2} + \log \alpha_i^{(j)} \right). \tag{4.83}$$

Por lo tanto, $\log q^*(z_i) \propto \sum_{j=1}^{K} z_i^{(j)} \log(g_i^{(j)} \alpha_i^{(j)})$, donde:

$$\log g_i^{(j)} = \mathbb{E}_{x^{(j)}} \left[\log p(y_i|x^{(j)}, z_i = j) \right] + \sum_{(i,i')\in\varepsilon} h_{i'}^{(j)} \log \frac{\gamma_1}{\gamma_2}. \tag{4.84}$$

Esta es una distribución multinomial con constante de normalización $\sum_{j=1}^{K} (\alpha_i^{(j)} g_i^{(j)})$.

Resumiendo

Lo visto hasta el momento nos conduce a las distribuciones:

$$\log q(x^{(j)}) = \sum_{i=1}^{m} h_i^{(j)} \log p(y_i|x^{(j)}, z_i = j) + \log p(x^{(j)}) - \log \mathcal{Z}_q^{(j)}, \tag{4.85}$$

$$\log q(z_i) = \sum_{j=1}^{K} z_i^{(j)} \log h_i^{(j)}, \tag{4.86}$$

donde $\mathcal{Z}_q^{(j)}$ es una constante de normalización y $h_i^{(j)}$ son los parámetros variacionales:

$$h_i^{(j)} = \mathbb{E}_{z_i} \left[z_i^{(j)} \right] = \frac{\alpha_i^{(j)} g_i^{(j)}}{\sum_{k=1}^{K} \alpha_i^{(k)} g_i^{(j)}}, \tag{4.87}$$

$$\log g_i^{(j)} = \mathbb{E}_{x^{(j)}} \left[\log p(y_i|x^{(j)}, z_i = j) \right] + \sum_{(i,i')\in\varepsilon} h_{i'}^{(j)} \log \frac{\gamma_1}{\gamma_2}, \tag{4.88}$$

y $\mathbb{E}_{x^{(j)}}$ y $\mathbb{E}_{z_i}$ son las esperanzas con respecto a $q(x^{(j)})$ y $q(z_i)$ respectivamente.

Los parámetros variacionales $\{h_i^{(j)}\}$ que aparecen en $q(z_i)$ y $q(x^{(j)})$, representan la dependencia entre X y Z (figura 4.2). $\{h_i^{(j)}\}$ es la probabilidad a

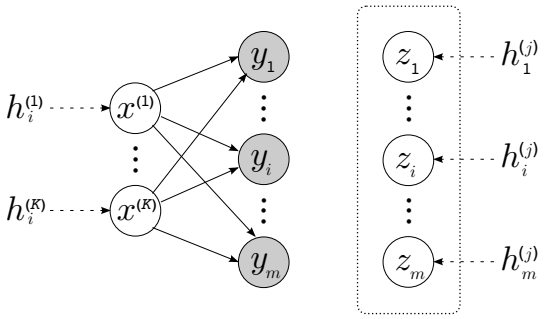

Figura 4.2: Modelo gráfico para la aproximación variacional para capas de texturas dinámicas. Las influencias de los parámetros variacionales están señalas con flechas punteadas.

posteriori de asignar el píxel y_i a la capa j y se estima con el logaritmo de la probabilidad esperada de asignar el píxel y_i a la capa j, con el estímulo adicional de $\log \frac{\gamma_1}{\gamma_2}$ por cada píxel vecino que también es asignado a la capa j. $\{h_i^{(j)}\}$ también mide la contribución de cada píxel y_i al factor $q(x^{(j)})$, que actúa como una asignación suave del píxel y_i a la capa j. También se puede ver en (4.85), que $\{h_i^{(j)}\}$ puede ser absorbido en $p(y_i|x^{(j)}, z_i = j)$, haciendo que $q(x^{(j)})$ sea la distribución de un sistema dinámico lineal parametrizado por $\hat{\Theta}_j = \{A^{(j)}, Q^{(j)}, C^{(j)}, R^{(j)}, \xi^{(j)}\}$ donde $R^{(j)}$ es una matriz diagonal con valores $[\frac{r^{(j)}}{h_1^{(j)}}, \ldots, \frac{r^{(j)}}{h_m^{(j)}}]$. Finalmente, $\log g_i^{(j)}$ es calculado reescribiendo (4.88) como:

$$\log g_i^{(j)} = \mathbb{E}_{x^{(j)}} \left[\frac{-1}{2r^{(j)}} \sum_{t=1}^{\tau} \|y_{i,t} - C_i^{(j)} x_t^{(j)}\|^2 - \frac{\tau}{2} \log 2\pi r^{(j)} \right] \tag{4.89}$$

$$+ \sum_{(i,i')\in\varepsilon} h_{i'}^{(j)} \log \frac{\gamma_1}{\gamma_2}$$

$$= \frac{-1}{2r^{(j)}} \left(\sum_{t=1}^{\tau} {y_{i,t}}^2 - 2C_i^{(j)} \sum_{t=1}^{\tau} \mathbb{E}_{x^{(j)}} \left[x_t^{(j)} \right] y_{i,t} \right. \tag{4.90}$$

$$\left. + C_i^{(j)} \sum_{t=1}^{\tau} \mathbb{E}_{x^{(j)}} \left[x_t^{(j)} {x_t^{(j)}}^T \right] \right) - \frac{\tau}{2} \log 2\pi r^{(j)}$$

$$+ \sum_{(i,i')\in\varepsilon} h_{i'}^{(j)} \log \frac{\gamma_1}{\gamma_2}$$

donde las esperanzas de $\mathbb{E}_{x^{(j)}}[x_t^{(j)}]$ y $\mathbb{E}_{x^{(j)}}[x_t^{(j)} {x_t^{(j)}}^T]$ son calculadas con el filtro Kalman de suavizado de A.3 para un sistema dinámico lineal con parámetros $\hat{\Theta}_j$.

El $q^*(X, Z)$ óptimo es encontrado iterando sobre cada píxel i, recalculando los parámetros variacionales $h_i^{(j)}$ de acuerdo a (4.87) y (4.88) hasta converger. Esto puede ser caro computacionalmente, porque requiere utilizar el filtro Kalman de suavizado para cada píxel. La carga puede ser reducida actualizando lotes de parámetros cada vez. En este trabajo se define un lote $\mathcal{B}$ como un

conjunto de nodos del campo aleatorio markoviano cuya vecindad no se solapa ([Bes74]), $\mathcal{B} = \{i | (i, i') \notin \varepsilon, \forall i' \in \mathcal{B}\}$. En la práctica, la actualización por lote, típicamente, converge a la solución alcanzada actualizando en serie, pero significativamente más rápido.

La aproximación variacional usando actualización por lote (sincrónica) se puede ver en el algoritmo (3):

Algorithm 3 Aproximación variacional para capas de texturas dinámicas

Require: Parámetros Θ, lotes $\{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$.
  Inicializar $\{h_i^{(j)}\}$.
  repeat
    {Recalcular los parámetros variacionales para cada lote}
    for $\mathcal{B} \in \{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$ **do**
      Calcular $\mathbb{E}_{x^{(j)}}[x_t^{(j)}]$ y $\mathbb{E}_{x^{(j)}}[x_t^{(j)} {x_t^{(j)}}^T]$ utilizando el filtro Kalman de suavizado con parámetros $\hat{\Theta}_j$, para $j \in \{1, \ldots, K\}$.
      for $i \in \mathcal{B}$ **do**
        Calcular $\log g_i^{(j)}$ usando (4.90), para $j \in \{1, \ldots, K\}$.
        Calcular $h_i^{(j)}$ usando (4.87), para $j \in \{1, \ldots, K\}$.
      end for
    end for
  until $h_i^{(j)}$converge

4.4.2. Inferencia aproximada

En lo que resta de la sección se discute la inferencia con la aproximación a posteriori de $q^*(X, Z)$.

Etapa E

En (4.19), las esperanzas con respecto a $p(X|Y)$ y $p(Z|Y)$ puede ser estimadas como:

$$\begin{array}{rclcrcl} \hat{x}_t^{(j)} & \approx & \mathbb{E}_{x^{(j)}}[x_t^{(j)}], & & \hat{P}_{t,t}^{(j)} & \approx & \mathbb{E}_{x^{(j)}}[x_t^{(j)} {x_t^{(j)}}^T], \\ \hat{z}_i^{(j)} & \approx & h_i^{(j)}, & & \hat{P}_{t,t-1}^{(j)} & \approx & \mathbb{E}_{x^{(j)}}[x_t^{(j)} {x_{t-1}^{(j)}}^T], \end{array} \tag{4.91}$$

donde $\mathbb{E}_{x^{(j)}}$ es la esperanza con respecto a $q^*(x^{(j)})$. Las esperanzas restantes de (4.19) son con respecto a $p(X|Y, z_i = j)$, y pueden ser aproximadas con $q^*(X|z_i = j)$ utilizando el algoritmo variacional condicionado $h_i^{(j)}$ binario, definido para forzar $z_i = j$. Tener en cuenta que si m es grande (como lo es para videos), fijar el valor de un solo $z_i = j$ tendrá un efecto insignificante a posteriori, dada la evidencia combinada del gran número de otros píxeles en la capa. Por eso, las esperanzas con respecto a $p(X|Y, z_i = j)$ pueden ser aproximadas con $q^*(X)$ cuando m es grande:

$$\begin{array}{rclcrcl} \hat{x}_{t|i}^{(j)} & \approx & \mathbb{E}_{x^{(j)}|z_i=j}[x_t^{(j)}] & \approx & \mathbb{E}_{x^{(j)}}[x_t^{(j)}], \\ \hat{P}_{t,t|i}^{(j)} & \approx & \mathbb{E}_{x^{(j)}|z_i=j}[x_t^{(j)} {x_t^{(j)}}^T] & \approx & \mathbb{E}_{x^{(j)}}[x_t^{(j)} {x_t^{(j)}}^T], \end{array} \tag{4.92}$$

donde $\mathbb{E}_{x^{(j)}|z_i=j}$ es la esperanza con respecto a $q^*(x^{(j)}|z_i = j)$.

Cota inferior de $p(Y)$

La convergencia es monitoreada con una cota inferior de $p(Y)$, que se obtiene a partir de la no negatividad de la divergencia KL y de (4.49):

$$\begin{aligned} \mathrm{KL}(q(X,Y)\|p(X,Z|Y)) = \mathcal{L}(q(X,Z)) + \log(p(Y)) \geq 0 \\ \Rightarrow \log(p(Y)) \geq -\mathcal{L}(q(X,Z)). \end{aligned} \tag{4.93}$$

Evaluando $\mathcal{L}$ para el q^* óptimo, la cota inferior es:

$$\begin{aligned} \log(p(Y)) \geq \sum_{j=1}^{K} \log \mathcal{Z}_q^{(j)} - \sum_{i=1}^{m}\sum_{j=1}^{K} h_i^{(j)} \log \frac{h_i^{(j)}}{\alpha_i^{(j)}} \\ + \sum_{(i,i')\in\varepsilon} \left(\log\gamma_2 + \sum_{j=1}^{K} h_i^{(j)} h_{i'}^{(j)} \log\frac{\gamma_1}{\gamma_2} \right) - \log \mathcal{Z}_Z \end{aligned} \tag{4.94}$$

Asignación de capa a posteriori máxima

Dado el vídeo Y observado, la asignación de capa a posteriori máxima Z (la segmentación) es:

$$Z^* = \arg\max_Z p(Z|Y) \tag{4.95}$$

$$= \arg\max_Z \int p(X,Z|Y)dX \tag{4.96}$$

$$\approx \int q^*(X,Z)dX \tag{4.97}$$

$$= \arg\max_Z \int \prod_{j=1}^{K} q^*(x^{(j)}) \prod_{i=1}^{m} q^*(z_i)dX \tag{4.98}$$

$$= \arg\max_Z \prod_{i=1}^{m} q^*(z_i). \tag{4.99}$$

Por lo tanto, la solución a posteriori máxima para Z es aproximada por las soluciones individuales a posteriori máximas para z_i:

$$z_i^* \approx \arg\max_j h_i^{(j)}, \forall i. \tag{4.100}$$

4.5. Inicialización

El algoritmo EM devuelve una solución óptima local y esta depende en gran medida de la inicialización de los parámetros del modelo. En la mayoría de los casos, la aproximación de la etapa de esperanza requiere también una estimación inicial de las asignaciones de capa $\hat{z}_i^{(j)}$. Si una segmentación inicial esta disponible, ambos problemas pueden ser resueltos de manera sencilla: los parámetros pueden ser aprendidos utilizando el método para la estimación de parámetros de texturas dinámicas 2.2 para cada región y la mascara de la segmentación como $\hat{z}_i^{(j)}$ inicial. En caso de que no haya una segmentación inicial disponible, se puede utilizar el método de mixturas de texturas dinámicas 3 para obtener una.

4.6. Segmentación

El video es segmentado asignando cada píxel a la capa más probable (condicionada sobre el video observado):

$$z_i^* = \arg\arg\max_j p(z_i = j|Y) \tag{4.101}$$

Este resultado es calculado durante la etapa E del algoritmo EM.

Capítulo 5

Resultados

En este capítulo se describen los experimentos realizados para poner a prueba la eficacia de los algoritmos de segmentación desarrollados en el presente trabajo.

Se realizaron experimentos con tres clases de videos. La primera clase incluye videos sintéticos que presentan movimiento circular con diferentes velocidades y texturas; la segunda, videos de fenómenos naturales estacionarios (agua, fuego, vegetación, etc.), que fueron combinados utilizando distintas máscaras; la tercera clase está compuesta por videos reales, sin ningún tipo de alteración. Las primeras dos clases contienen secuencias con $K = 2, 3, 4$ regiones de diferentes texturas de video; mientras que la tercera contiene videos reales varios (tráfico de automóviles, personas caminando, etc). Antes de ser utilizados los videos de color fueron convertidos a escala de grises.

Para reducir la memoria y los cálculos necesarios, se asumió que $\bar{y}_i$ puede ser estimado por la media empírica del video observado, o sea, $\bar{y}_i \approx \frac{1}{\tau} \sum_{t=1}^{\tau} y_{i,t}$. Esto es válido mientras τ sea "grande" y A sea estable, suposiciones razonables para video estacionario. Dado que la media empírica es fija para un Y dado, se puede sustraer la media empírica del video y definir $\bar{y}_i = 0$.

Las medidas que se utilizaron para evaluar la eficiencia de los algoritmos son rand index y tiempo de ejecución. El rand index es una medida de similitud entre dos agrupaciones de datos. Intuitivamente, es la probabilidad de coincidencia entre dos agrupaciones y se utiliza para comparar el resultado de un algoritmo con la agrupación ideal. El tiempo de ejecución es la cantidad de tiempo que demoró el algoritmo en devolver la agrupación.

Dado que el rendimiento del algoritmo de texturas dinámicas para $K = 2$ capas fue el más pobre de los tres, no se implementó la extensión del mismo para $K > 2$ capas.

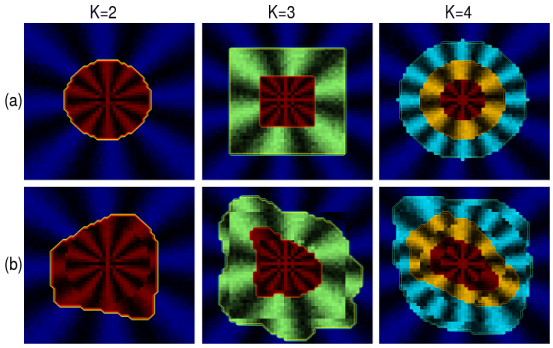

Figura 5.1: Videos de movimiento circular. (a) Contorno ideal. (b) Contorno inicial.

5.1. Videos con Movimiento circular

Estos experimentos fueron realizados sobre videos conteniendo diferentes anillos con distinto movimiento circular (Figura 5.1). Cada secuencia $\mathcal{I}_{x,y,t}$ tiene dimensión 50×50, y fue generada de acuerdo a:

$$\mathcal{I}_{x,y,t} = 128 \cos\left(c_r\theta + \frac{2\pi}{T_r}t + v_t\right) + 128 + w_t, \tag{5.1}$$

donde $\theta = \arctan(\frac{x-25}{y-25})$ es el ángulo del píxel (x, y) relativo al centro del cuadro del video, $v_t \sim \mathcal{N}(0, (2\pi/50)^2)$ es el ruido de la fase, y $w_t \sim \mathcal{N}(0, (10)^2)$ es el ruido de la observación. El parámetro $T_r \in \{5, 10, 20, 40\}$ determina la velocidad de cada anillo, mientras que c_r determina el número de veces que se repite la textura alrededor del anillo. Se eligió c_r de forma que todos los anillos de texturas tuvieran el mismo período espacial. Se generaron secuencias para $K = \{2, 3, 4\}$ anillos sobre máscaras circulares y cuadradas.

Algunos resultados de la segmentación se pueden observar en la figura 5.2. Fueron obtenidos utilizando MTD y CTD para un mismo contorno inicial fijo. Tanto las texturas dinámicas como las mixturas de texturas dinámicas tienden a segmentaciones incorrectas basadas en la dirección local del movimiento. En algunos casos, además, se asignó incorrectamente un segmento de los bordes entre los anillos, mostrando como el bajo rendimiento en los bordes de segmentación basada en parches puede generar importantes problemas. Por otro lado, las capas de texturas dinámicas segmentaron de forma correcta todos los anillos sin importar su forma, favoreciendo la homogeneidad global sobre los grupos localizados de segmentos orientados. Tanto las texturas dinámicas como las mixturas de texturas dinámicas fallaron al exhibir esta propiedad, lo que se ve reflejado en los resultados del rand index promedio obtenido por las capas de las texturas dinámicas:

Método	$K = 2$	$K = 3$	$K = 4$
CTD	1.0000 (2)	1.0000 (2)	1.0000 (2)
MTD	0.6727 (10)	0.4305 (15)	0.3543 (10)
TD	0.6150 (15)	-	-

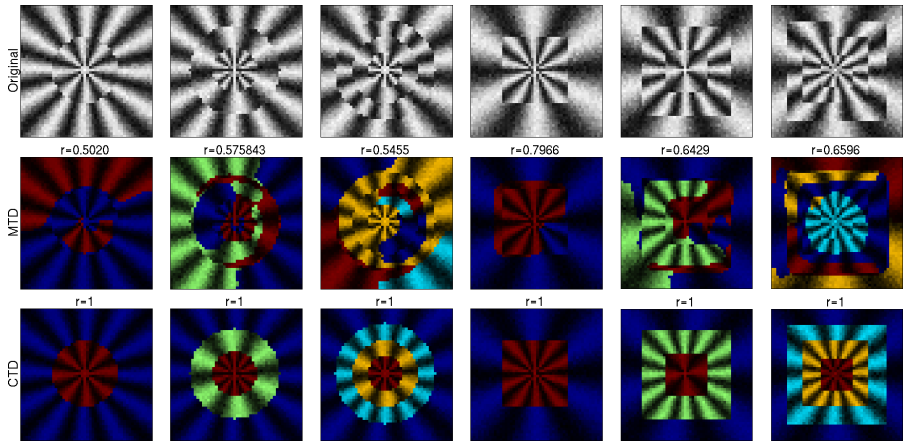

Figura 5.2: Segmentación de movimiento circular utilizando MTD y CTD.

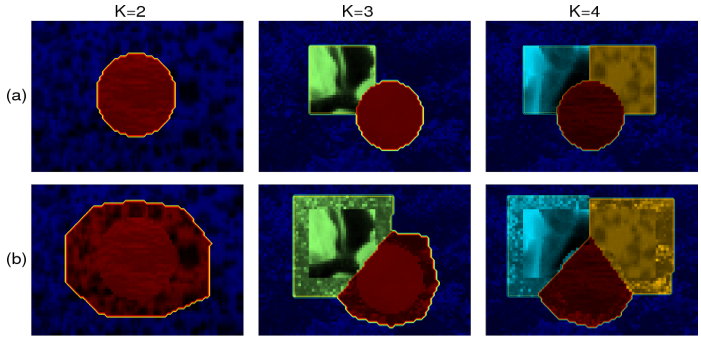

Figura 5.3: Videos de fenómenos compuestos. (a) Contorno ideal. (b) Contorno inicial.

5.2. Videos de Fenómenos compuestos

Aquí se evalúan los resultados sobre videos compuestos de diferentes texturas de la base de datos [CV08]. La base contiene 299 secuencias con $K = \{2, 3, 4\}$ regiones de diferentes texturas de video. Cada método fue ejecutado con el siguiente conjunto de parámetros:

Método	Estado (n)	T. Ventana	Orden Vec.	Cohesión Vec. (γ)
CTD	{5,10,15,20}	-	{1,2,3,4}	{3,5,7}
MTD	{5,10,15,20}	{5,7,10,15}	-	-
TD	{5,10,15,20}	{5,7,10,15}	-	-

Se utilizó un mismo contorno inicial fijo (Figura 5.3). El rand index promedio fue calculado para cada K. No se realizó ningún procesamiento posterior.

La siguiente tabla muestra el rendimiento obtenido, con el mejor n, por cada algoritmo:

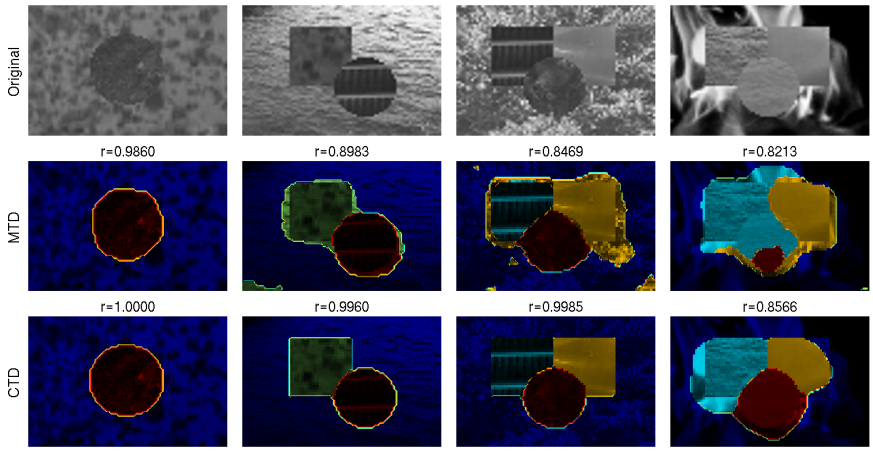

Figura 5.4: Mejores resultados de MTD frente a CTD.

Método	$K = 2$	$K = 3$	$K = 4$
CTD	0.9173 (20)	0.9172 (20)	0.9260 (15)
MTD	0.9092 (20)	0.8322 (15)	0.8145 (15)
TD	0.7756 (15)	-	-

CTD fue el algoritmo de mejor rendimiento, seguido de MTD. En la figura 5.4 se pueden observar los errores que generó el algoritmo de MTD. Estos errores se produjeron sobre todo en los bordes, porque el algoritmo de mixturas funciona sobre "parches" (matriz de píxeles) para representar cada componente. En cambio, el algoritmo de CTD trabaja a nivel de píxel, lo que permite un mayor nivel de detalle. En los resultados obtenidos las CTD presentan un rendimiento superior, salvo en el último caso. Éste contiene una textura de fuego como fondo, que presenta un comportamiento heterogéneo a través del tiempo. Esto hizo que el algoritmo confunda esta textura con las otras dos, generando una segmentación de menor calidad.

La figura 5.5 muestra el rendimiento de los algoritmos frente al tamaño del vector de estados n para $K = \{2, 3, 4\}$, evidenciando que la segmentación con CTD y MTD responde adecuadamente a la selección de n.

En cuanto a los tiempos de respuesta, (figura 5.6), el mayor pertenece a CTD. Para MTD y TD el tiempo obtenido fue prácticamente el mismo. Para $K = 4$, los tiempos de CTD y MTD se acercan.

Finalmente, se examinó el rendimiento de los algoritmos frente al resto de los parámetros (figura 5.7). Se puede observar que tanto el orden de la vecindad como el tamaño de la ventana inciden sobre el rendimiento CTD y MTD/TD respectivamente. No así la cohesión de la vecindad γ, que muestra prácticamente los mismos resultados.

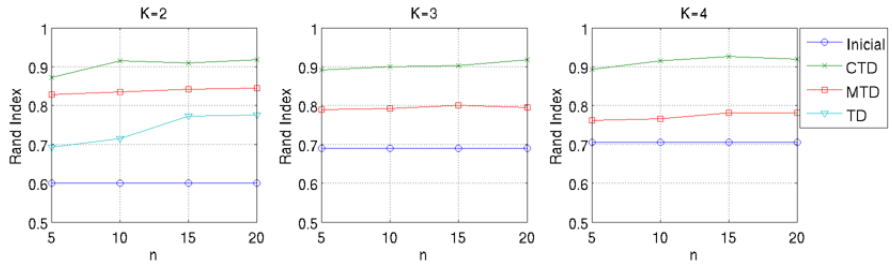

Figura 5.5: Rand index frente a n para videos con $K = \{2, 3, 4\}$ segmentos.

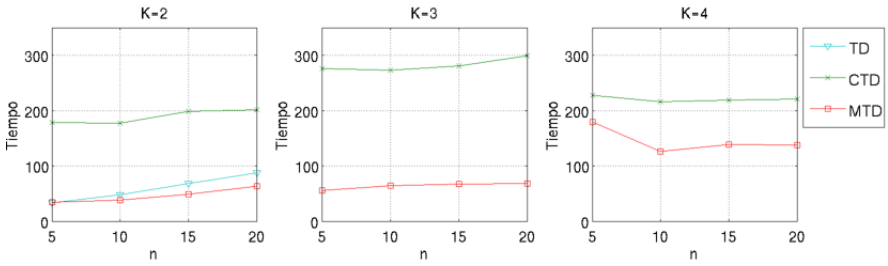

Figura 5.6: Tiempo de respuesta frente a n para videos con $K = \{2, 3, 4\}$ segmentos.

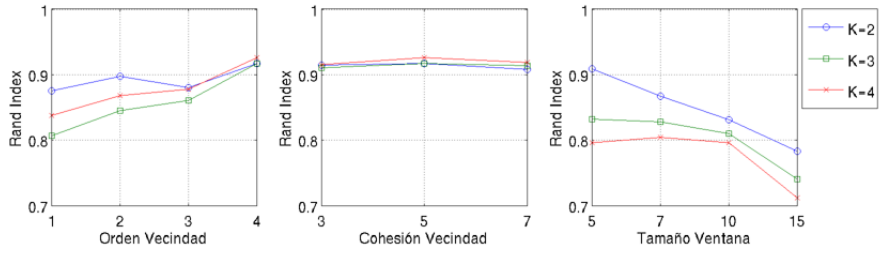

Figura 5.7: Rendimiento de la segmentación frente al orden de la vecindad y cohesión de la vecindad para CTD, y tamaño de ventana para MTD y TD.

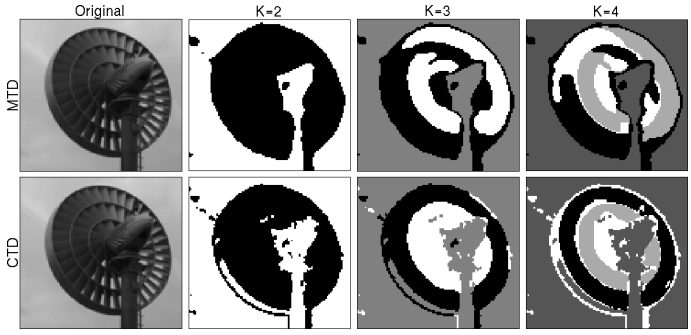

Figura 5.8: Turbina de viento segmentada con $K = \{2, 3, 4\}$ regiones utilizando MTD y CTD para $n = 2$.

5.3. Videos Reales

A continuación se presentan los resultados de las pruebas de segmentación realizadas sobre secuencias de video reales. Estos fueron obtenidos de la base de datos UCF [PFH] y de www.youtube.com. En todos los casos se utilizó un campo aleatorio markoviano de segundo orden, una cohesión de vecindad $\gamma = -\gamma = 5$, y un vector de estados n correspondiente al mejor resultado obtenido para cada segmentación.

Dado que no se tiene un contorno inicial para estos videos, para realizar cada prueba primero se ejecutó el algoritmo MTD con el método de inicialización de división de componente 3.3.3, y luego se utilizó el resultado de esta segmentación como contorno inicial del algoritmo CTD.

La figura 5.8 muestra la segmentación de una turbina de viento utilizando MTD y CTD para $K = \{2, 3, 4\}$. Para $K = 2$, tanto MTD como CTD, separaronn el fondo estático del movimiento de la turbina. Sin embargo, para $K = \{3, 4\}$ la segmentación del movimiento de los anillos de la turbina obtenido por CTD no se replicó por parte de MTD. En su lugar, MTD segmentó el video en regiones de acuerdo a la dirección dominante del movimiento local (hacia arriba o hacia abajo). Este problema es idéntico al encontrado para las secuencias sintéticas de la figura 5.2: la incapacidad para tratar la homogeneidad global cuando cuando el video es heterogéneo localmente. Por otro lado, la preferencia de las CTD por regiones de tamaños muy diferentes ilustra su robustez a este tipo de problema. La fuerte heterogeneidad local del flujo óptico en las regiones de los anillos es bien explicada por la homogeneidad global de las correspondientes capas dinámicas. Y aunque para $K = 3$ los anillos interiores quedan en una misma capa, esto no sucede para $K = 4$. Otro ejemplo de este fenómeno fue obtenido en la figura 5.9. El torbellino contiene diferentes niveles de movimiento y turbulencia de agua que CTD segmentó de forma correcta en diferentes regiones globalmente homogéneas, no así MTD.

Otro problema de las MTD es el tratamiento de las zonas de borde. En la figura 5.10 se pueden apreciar segmentaciones de tráfico en autopista circulando

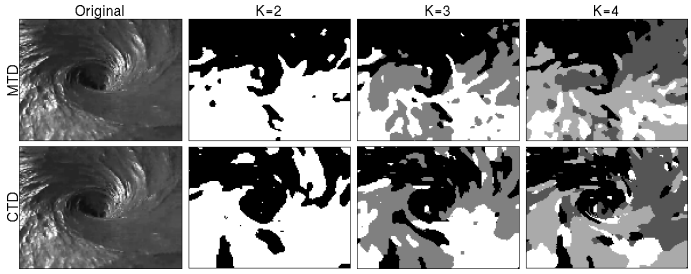

Figura 5.9: Remolino de agua segmentado con $K = \{2, 3, 4\}$ regiones utilizando MTD y CTD para $n = 2$.

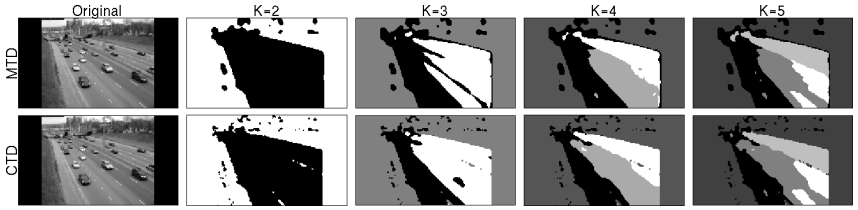

Figura 5.10: Tráfico de automóviles en una autopista segmentado con $K = \{2, 3, 4, 5\}$ regiones utilizando MTD y CTD para $n = 10$.

en dos direcciones y a diferentes velocidades. Hasta $K = 4$ se han diferenciado: el fondo estático, los automóviles que se alejan, los que se acercan por el carril izquierdo a poca velocidad y por el derecho a una mayor velocidad. Para $K = 5$ dentro del carril derecho aparece un nuevo segmento que tiene una mayor velocidad. Esto se debe a que la perspectiva del video genera ese efecto, aunque la velocidad de los automóviles sea la misma. Tanto MTD como CTD obtuvieron para todos los valores de K un rendimiento similar, si no se tienen en cuenta los bordes. Los borden presentan un problema para las MTD, ya que no tienen una noción de espacialidad global, de cómo están distribuidos los píxeles dentro del video. Las CTP, en cambio, poseen el campo aleatorio markoviano que incide sobre la probabilidad de pertenencia del píxel a una determinada capa en base a su vecindad, lo que genera resultados precisos. Otro ejemplo puede observarse en la figura 5.11, donde se presenta una casa bajo un cielo estrellado. Nuevamente, a grandes rasgos los resultados que se obtuvieron con ambos métodos son similares, pero queda claro que CTD fue superior en el nivel de detalle.

Por último, en la figura 5.12 se presenta un resultado que muestra un problema del método de inicialización. El video muestra una cascada que cae sobre un estanque. El agua de la cascada, al entrar en contacto con el agua del estanque, desprende partículas finas que forma una especie de "niebla". Sería esperable que estos tres fenómenos, más el fondo estático, queden en segmentos diferentes. Para $K = 2$ se separa el fondo del resto, para $K = 3$ además del fondo se diferencia la cascada, pero para $K = 4$ se separa a la cascada en dos, el flujo interno del externo, y no al estanque de la niebla. Esto sucede porque el ruido del componente visual de la cascada es el más importante y fue elegido para

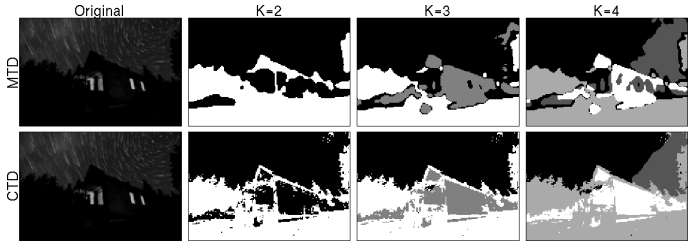

Figura 5.11: Recorrido de estrellas segmentado con $K = \{2, 3, 4\}$ regiones utilizando MTD y CTD para $n = 10$.

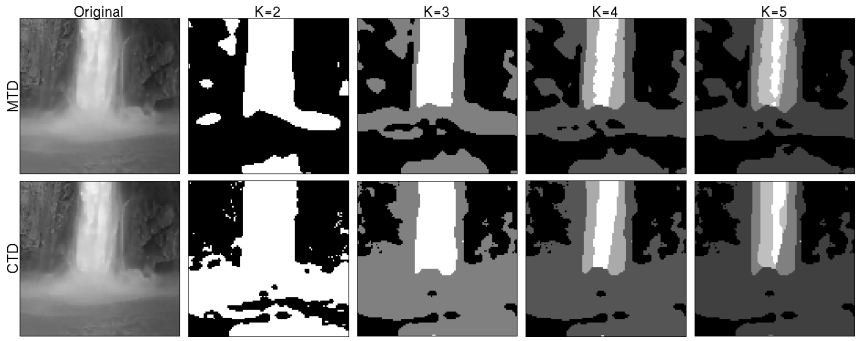

Figura 5.12: Cascada segmentada con $K = \{2, 3, 4, 5\}$ regiones utilizando MTD y CTD para $n = 10$.

separar y generar una nueva componente. Inclusive para $K = 5$, el fenómeno se repitió.

Capítulo 6

Conclusiones

Este trabajo ha analizado la clase de procesos dinámicos visuales que exhiben estacionalidad temporal y se refiere a secuencias de video de esta clase como *texturas dinámicas*. Se busca el modelo más simple que pueda capturar, al menos, las estadísticas de segundo orden de la apariencia en espacio-tiempo y la variabilidad de la figura. Para esto se utiliza análisis estadístico lineal que permite derivar procedimientos de aprendizaje muy eficientes en forma cerrada. Se muestra experimentalmente que aún las instancias más simples del modelo permiten capturar la esencia de la información contenida en una gran clase de texturas dinámicas. Modelando la apariencia y la variabilidad de la figura se obtiene un modelo lineal condicional, que es globalmente no lineal, y captura estadísticas temporales de un orden superior de secuencias de video.

Se analizó el significado de los parámetros del modelo de texturas dinámicas y se encontró que mediante su manipulación cuidadosa es posible cambiar la apariencia y variar la velocidad, lo que puede ser útil para animación de video. Por medio de la comparación de los subespacios de todas las realizaciones que pueden ser generadas por un sistema dinámico lineal se ha demostrado que se puede discriminar entre modelos. Es posible, por lo tanto, generar una base de datos de modelos que represente el conocimiento a priori y utilizar este resultado para construir un sistema de reconocimiento basado en la regla *knn* (los k vecinos más cercanos). También se utilizó este mecanismo para construir un sistema de segmentación basado en regiones que extiende Mumford-Shah para segmentar las estadísticas espacio-temporales de las secuencias de video.

Asimismo se han generado dos extensiones del modelo de texturas dinámicas: las mixturas de texturas dinámicas y las capas de texturas dinámicas. Estas extensiones agregan una variable oculta adicional para sirve para modelar la coexistencia de varias texturas dinámicas dentro de un mismo esquema. Se han derivado los algoritmos EM para la estimación de los parámetros del modelo de máxima-verosimilitud utilizando secuencias de video de entrenamiento. Para capas de texturas dinámicas, se han analizado dos alternativas para inferir las variables ocultas x y z, dado que la distribución a posteriori es computacionalmente intratable: un generador de muestras de Gibbs y una aproximación variacional eficiente. Los dos algoritmos de inferencia aproximada producen resultados comparables, aunque el método de Gibbs superó a la aproximación

variacional a costa de un mayor tiempo de respuesta.

También se realizaron experimentos extensivos, con secuencias compuestas de texturas reales y secuencias de video reales que pusieron a prueba la habilidad de los modelos (y algoritmos) propuestos para segmentar video en regiones de apariencia y comportamiento coherente. La combinación de las capas de texturas dinámicas y la inferencia variacional arrojó un resultado superior al resto de los métodos evaluados, en tiempo y calidad. Este método ha generado también segmentaciones con mejor localización espacial que las texturas dinámicas y mixturas de texturas dinámicas, con representaciones localizadas. Finalmente, ha sido demostrada la robustez del modelo para segmentar secuencias de video reales que descifran diferentes clases de escenas.

Más allá que los resultados obtenidos fueron realmente muy buenos, quedan pendiendientes algunos temas referidos al rendimiento temporal, que pueden ser mejorados. El foco primario a tener en cuenta es el filtro de Kalman, ya que éste es el núcleo de la etapa de esperanza del algoritmo EM para estimar x, tanto para mixturas como para capas de texturas dinámicas. Una alternativa para alivianar su tarea sería sub-muestrear la entrada antes de ejecutarlo. De esta manera, si la cantidad de muestras es grande, se podría tomar un número de muestras inferior que sea representativo de la totalidad de las muestras, sin afectar la calidad de la estimación. También sería interesante analizar variantes del algoritmo que lo hagan más eficiente bajo alguna condición del modelo, sin que ello afecte el rendimiento general del mismo, o bien, analizar alternativas para la estimacion del estado oculto x sin la utilización de Kalman.

Bajo el modelo de capas de texturas dinámicas utilizando inferencia por aproximación variacional, se observa un problema cuando se realiza la estimacíon de la variable oculta z. En cada iteración dentro de la etapa de esperanza se intenta mejorar la asignación de cada píxel a una capa, analizando tanto su probabilidad de pertenencia como su vecindad. Lo que se observa es que, si el píxel fue asignado a una capa en la primera iteración, las iteraciones posteriores arrojarán el mismo resultado si su vecindad no es alterada. Esto es algo que sucede muy a menudo en grandes regiones del video, lo que debería mejorar el rendimiento de iteraciones posteriores a la primera.

Apéndices

Apéndice A

Conceptos

A.1. Norma Frobenius

La norma euclídea en $\mathbb{R}^{m\times n}$ es

$$||A|| = \left(\sum_{i=1}^{m}\sum_{i=1}^{n} \|a_{ij}\|^2\right)^{\frac{1}{2}} \tag{A.1}$$

Esta norma también es llamada norma Frobenius y se suele representar por $||A||_F$. En algunos sitios se le llama también norma de Schur o de Hilbert-Schmidt.

La norma Frobenius se puede escribir alternativamente de cualquiera de las dos formas siguientes:

$$\text{a) } ||A||_F = \text{tr}\,(A^*A)^{\frac{1}{2}} \tag{A.2}$$

$$\text{b) } ||A||_F = \left(\sum_{i=1}^{\text{mín}\{m,n\}} \sigma_i^2\right)^{\frac{1}{2}} \tag{A.3}$$

donde A^* denota la transpuesta conjugada de A, σ_i son los valores singulares de A y "tr" es la traza.

La norma de Frobenius es submultiplicativa y es muy útil para álgebra lineal numérica. Esta norma es, en general, más fácil de calcular que las normas inducidas.

A.2. Algoritmo EM

En estadística, un algoritmo expectation-maximization (EM) [DLR77] es un método utilizado para encontrar los parámetros de máxima verosimilitud o máximos a posteriori (MAP) en modelos estadísticos, donde el modelo depende de variables ocultas. EM es un método iterativo que alterna entre una etapa de esperanza (E), que calcula la esperanza del logaritmo de la verosimilitud usando las estimaciones actuales, y una etapa de maximización (M), que calcula los

parámetros que maximizan el valor esperado de la etapa E. Estas estimaciones son usadas luego para determinar las variables ocultas en la próxima etapa E.

Formalmente, dado un modelo estadístico formado por un conjunto Y de información observada, un conjunto X de información oculta y un vector de parámetros desconocidos Θ, junto con una función de verosimilitud $L(\Theta; X, Y) = p(X, Y|\Theta)$, el estimador de máxima verosimilitud (EMV) de los parámetros desconocidos está determinando la verosimilitud marginal de la información observada:

$$L(\Theta; Y) = p(Y|\Theta) = \sum_X p(X, Y|\Theta) \tag{A.4}$$

Sin embargo, este cálculo es generalmente intratable. El algoritmo EM busca encontrar el EMV de la verosimilitud marginal iterativamente aplicando los siguientes dos pasos:

Etapa E: Calcula la esperanza del logaritmo de la función de verosimilitud con respecto a la distribución condicional de X dado Y utilizando la estimación actual de los parámetros $\Theta^{(t)}$:

$$Q(\Theta|\Theta^{(t)}) = \mathbb{E}_{X|Y,\Theta^{(t)}}[\log L(\Theta; X, Y)] \tag{A.5}$$

Etapa M: Calcula los parámetros que maximizan la esperanza hallada del calculo anterior:

$$\Theta^{(t+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(t)}) \tag{A.6}$$

En este trabajo se utiliza el filtro Kalman de suavizado para calcular la esperanza de la etapa E; para obtener los parámetros en la etapa M se deriva el logaritmo de la función de verosimilitud, obteniendo sus máximos.

A.3. Filtro Kalman

El filtro Kalman de suavizado [SS82] estima la media y la covarianza del estado x_t de un sistema dinámico lineal, condicionado sobre toda la secuencia observada $\{y_1, \ldots, y_\tau\}$. También puede ser utilizado para calcular el logaritmo de la verosimilitud de la secuencia observada de forma eficiente.

Definidas las esperanzas condicionales sobre la secuencia observada del instante $t = 1$ al $t = s$ como

$$\hat{x}_t^s = \mathbb{E}_{x|y_1,\ldots,y_s}(x_t), \tag{A.7}$$

$$\hat{V}_t^s = \mathbb{E}_{x|y_1,\ldots,y_s}\left((x_t - \hat{x}_t^s)(x_t - \hat{x}_t^s)^T\right), \tag{A.8}$$

$$\hat{V}_{t,t-1}^s = \mathbb{E}_{x|y_1,\ldots,y_s}\left((x_t - \hat{x}_t^s)(x_{t-1} - \hat{x}_{t-1}^s)^T\right), \tag{A.9}$$

luego la media y la covarianza condicionadas sobre toda la secuencia observada son $\hat{x}_t^\tau$, $\hat{V}_t^\tau$ y $\hat{V}_{t,t-1}^\tau$. Los estimados son calculados utilizando una serie de

ecuaciones recursivas, para $t = 1, \ldots, \tau$:

$$\hat{V}_t^{t-1} = A\hat{V}_{t-1}^{t-1}A^T + Q, \tag{A.10}$$

$$K_t = \hat{V}_t^{t-1}C^T(C\hat{V}_t^{t-1}C^T + R)^{-1}, \tag{A.11}$$

$$\hat{V}_t^t = \hat{V}_t^{t-1} - K_tC\hat{V}_t^{t-1}, \tag{A.12}$$

$$\hat{x}_t^{t-1} = A\hat{x}_{t-1}^{t-1}, \tag{A.13}$$

$$\hat{x}_t^t = \hat{x}_t^{t-1} + K_t(y_t - C\hat{x}_t^{t-1}), \tag{A.14}$$

donde las condiciones iniciales son $\hat{x}_1^0 = \mu$ y $\hat{V}_1^0 = S$. Las estimaciones $\hat{x}_t^\tau$ y $\hat{V}_t^\tau$ se obtienen por medio de recursiones inversas. O sea, para $t = \tau, \ldots, 1$

$$J_{t-1} = \hat{V}_{t-1}^{t-1}A^T(\hat{V}_t^{t-1})^{-1}, \tag{A.15}$$

$$\hat{x}_{t-1}^\tau = \hat{x}_{t-1}^{t-1} + J_{t-1}(\hat{x}_t^\tau - A\hat{x}_{t-1}^{t-1}), \tag{A.16}$$

$$\hat{V}_{t-1}^\tau = \hat{V}_{t-1}^{t-1} + J_{t-1}(\hat{V}_t^\tau - \hat{V}_t^{t-1})J_{t-1}^T. \tag{A.17}$$

La covarianza $\hat{V}_{t,t-1}^\tau$ se calcula recursivamente, para $t = \tau, \ldots, 2$

$$\hat{V}_{t-1,t-2}^\tau = \hat{V}_{t-1}^{t-1}J_{t-2}^T + J_{t-1}(\hat{V}_{t,t-1}^\tau - A\hat{V}_{t-1}^{t-1})J_{t-2}^T \tag{A.18}$$

con la condición inicial $\hat{V}_{t,t-1}^\tau = (I - K_\tau C)A\hat{V}_{t-1}^{t-1}$.

Adicionalmente, si R es i.i.d. o es una matriz de covarianza diagonal (por ejemplo $R = rI_m$), el filtro puede ser calculado utilizando el lema de la matriz inversa, que posee un mejor rendimiento:

$$(UCV + A)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \tag{A.19}$$

donde $A \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times k}$, $C \in \mathbb{R}^{k\times k}$ y $V \in \mathbb{R}^{k\times n}$. Esta igualdad es útil cuando A^{-1} ha sido calculado (o es fácil de calcular) y se desea calcular $(UCV + A)^{-1}$. Si C tiene una dimensión mucho menor que A, calcular la inversa de $C^{-1} + VA^{-1}U$ es mucho más eficiente que invertir $UCV + A$.

Aplicando el lema A.19 sobre la ecuación A.11 del filtro de Kalman, y asumiendo que $R = rI_m$, se obtiene:

$$K_t = \hat{V}_t^{t-1}C^T\left[R^{-1} - R^{-1}C(\hat{V}_t^{t-1^{-1}} + C^TR^{-1}C)^{-1}C^TR^{-1}\right] \tag{A.20}$$

$$= \hat{V}_t^{t-1}C^T\left[\frac{1}{r}I_m - \frac{1}{r^2}C(\hat{V}_t^{t-1^{-1}} + \frac{1}{r}C^TC)^{-1}C^T\right], \tag{A.21}$$

A.4. Kullback-Leibler

En teoría de la probabilidad la divergencia de Kullback-Leibler [Bis07] es un indicador asimétrico de la similitud entre dos funciones de distribución P y Q. Generalmente P representa la "verdadera" distribución de los datos, mientras que Q representa una teoría, modelo, descripción o aproximación de P.

La divergencia de Kullback-Leibler entre dos funciones de distribución P y Q de una variable aleatoria contínua, suele representarse así:

$$\mathrm{KL}(P\|Q) = \int_{-\infty}^{\infty} p(i) \log \frac{p(i)}{q(i)} di \tag{A.22}$$

Se trata de una divergencia y no una métrica por no ser simétrica, o sea:

$$\mathrm{KL}(P\|Q) \neq \mathrm{KL}(Q\|P) \tag{A.23}$$

En palabras, es el promedio de la diferencia logarítmica entre las probabilidades de P y Q, donde el promedio es tomado usando las probabilidades de P. Se utiliza comúnmente en lugar del muestreo de Gibbs, ya que éste es computacionalmente muy costoso.

A.5. Distancia de Mahalanobis

En estadística, la Distancia de Mahalanobis es una medida de distancia introducida por Mahalanobis. Su utilidad radica en que es una forma de determinar la "similitud" entre dos variables aleatorias multidimensionales. Se diferencia de la distancia euclídea en que tiene en cuenta la correlación entre las variables aleatorias.

Formalmente, la distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad $\vec{x}$ y $\vec{y}$ con matriz de covarianza Σ se define como:

$$d_m(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_\Sigma = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}. \tag{A.24}$$

Bibliografía

[Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

[Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.

[CCMM00] Katrien De Cock, Katrien De Cock, Bart De Moor, and Bart De Moor. Subspace angles between linear stochastic models. In *In Proc. the 39th IEEE Conference on Decision and Control*, pages 1561–6, 2000.

[CSV00] Tony F. Chan, B. Yezrielev Sandberg, and Luminita A. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11:130–141, 2000.

[CV08] Antoni B. Chan and Nuno Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):909–926, May 2008.

[CV09] Antoni B. Chan and Nuno Vasconcelos. Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1862–1879, 2009.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[Dor05] Gianfranco Doretto. *Dynamic textures: Modeling, learning, synthesis, animation, segmentation, and recognition.* ProQuest LLC, Ann Arbor, MI, 2005. Thesis (Ph.D.)–University of California, Los Angeles.

[GG87] Stuart Geman and Donald Geman. Readings in computer vision: issues, problems, principles, and paradigms. chapter Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, pages 564–584. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[Lju99] Lennart Ljung. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall, 2 edition, January 1999.

[Mar00] R. J. Martin. A metric for ARMA processes. *IEEE Transactions on Signal Processing*, 48(4):1164–1170, April 2000.

[PFH] Renaud Péteri, Sándor Fazekas, and Mark J. Huiskes. DynTex : a Comprehensive Database of Dynamic Textures. *Pattern Recognition Letters*, doi: 10.1016/j.patrec.2010.05.009. http://projects.cwi.nl/dyntex/.

[SS82] R H Shumway and D S Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[Wei00] Alan Weinstein. Almost invariant submanifolds for compact group actions. *Journal of the European Mathematical Society*, 2:53–86, 2000. 10.1007/s100970050014.