

Seguimiento del Cuerpo Humano Mediante Contornos Dinámicos

Tesis de Licenciatura

Alumnos:

Martín Sánchez 0019/96
msanchez@dc.uba.ar

Sergio Romano 246/96
sromano@dc.uba.ar

Director:

Marta Mejail
marta@dc.uba.ar

Codirector:

Julio Jacobo
jacobo@dc.uba.ar

Abril 2003



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

RESUMEN

El seguimiento de un objeto es identificar su posición y acompañar sus movimientos a lo largo del tiempo mediante una curva que lo contornea. Si además se considera el seguimiento sin markers, la tarea se convierte en algo más complejo.

Existen varios métodos para realizar seguimientos de objetos sin markers, entre ellos se encuentran los basados en modelos. Estos métodos contienen un modelo del objeto que se quiere seguir, el cual facilita y ayuda al sistema de seguimiento en su funcionamiento. Entre los esquemas de seguimiento basados en modelos, se encuentra el seguimiento por medio de Contornos Activos, el cual considera el uso de curvas que sufren transformaciones para adaptarse al contorno del objeto a seguir. Si se agrega a estos sistemas un modelo de estimación de movimiento se lo robustecerá, permitiéndole enfrentarse con diversos problemas tales como el ocultamiento del objeto que se sigue y el ruido presente en la escena. En este último caso, el sistema se denomina de Contornos Dinámicos.

El presente trabajo muestra una implementación de un método para seguimiento basado en Contornos Dinámicos. Se presenta una implementación que trabaja sobre un modelo simple a la vez que establece un equilibrio entre la velocidad de proceso, la calidad y el tipo de movimiento reconocido. Se muestra además una modificación del método de Contornos Dinámicos, la cual permite el seguimiento de objetos de una manera mucho más eficiente en tiempo y recursos del procesador. Se plantean algunas consideraciones para trabajar específicamente con secuencias de video del cuerpo humano. Algunas de ellas fueron implementadas y mostraron resultados satisfactorios.

El sistema presentado permite el aprendizaje de la dinámica del movimiento de los objetos seguidos, mejorando el seguimiento cuando se presentan objetos que se mueven y deforman de manera similar. Los resultados obtenidos por la ejecución del sistema en diversos videos de prueba han mostrado que el mismo se muestra robusto, aún en situaciones adversas, con ruido, ocultamiento y deformaciones. Asimismo, las pruebas realizadas, muestran que el sistema logra un buen funcionamiento en tiempo real, sin necesidad de contar con hardware costoso.

Aunque el sistema presentado aún puede ser mejorado en diversos aspectos, se cree que cumple, en gran parte, con los requerimientos necesarios para funcionar en gran cantidad de aplicaciones de identificación y seguimiento de personas. Entre estas áreas se pueden contar aplicaciones de medicina, seguridad, animación computada, etc.

ABSTRACT

Tracking an object is identifying its position and following its movements along the time, by the deformation of a spline who fits it. Without markers, the tracking becomes even more complex.

There are many methods for object tracking without markers and among them appears the model-based methods. This methods are coupled with a followed object model, which makes the tracking easier. The Active Contours method is a model-based method, which uses a spline template which is transformed in order hold the followed object contour. If a motion estimation model is added to this method, it will become more reliable and it will be able to solve out problems like occlusion and scene noise. In this case, the method is known as a Dynamic Contour one.

The work which is presented, shows a tracking system based in a Dynamic Contour method. It is presented an implementation that works with a simple model and states a balance between processing performance, tracking quality and quantity of recognizable movements. Besides, it is presented an addition to the Dynamic Contour method, which allows the tracking of objects using less time and resources. Some considerations to work with the human body are discussed, and some of them which have been implemented, show good results.

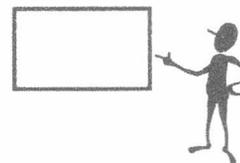
The system presented allows for the learning of the dynamics of the movement of the followed objects, improving the tracking when working over some other object with moves and deforms in a similar manner. The results obtained by the use of the system in a variety of sample videos, shows that the system is stronger even when working with occluded objects and noisy scenes. Moreover, the system shows a very good real-time performing, without the need for expensive hardware.

Even the presented system could be improved, it complies, mostly, with the requirements which makes it able to work in applications which require identifying a following people. Between the areas in which the system could be useful, you can find medicine applications, surveillance and computing animation.

Índice

CAPÍTULO 1: INTRODUCCIÓN	6
1.1 OBJETIVOS DEL PRESENTE TRABAJO	6
1.2 QUÉ SE ENTIENDE POR SEGUIMIENTO	6
1.3 TRABAJO PREVIO REALIZADO	6
1.4 QUÉ PROBLEMAS PUEDE PRESENTAR EL SEGUIMIENTO	10
1.5 QUÉ SE VERÁ EN LOS SIGUIENTES CAPÍTULOS	11
CAPÍTULO 2: SPLINES Y ESPACIO DE FORMAS.....	13
2.1 FUNCIONES B-SPLINE.....	14
2.1.1 <i>Funciones bases</i>	15
2.1.2 <i>Bases finitas</i>	16
2.1.3 <i>Norma y producto interno de funciones spline</i>	17
2.2 CURVAS PARAMÉTRICAS B-SPLINE	19
2.2.1 <i>Vector de control</i>	19
2.2.2 <i>Norma y producto interno de curvas</i>	20
2.2.3 <i>Reparametrización de curvas</i>	21
2.3 ESPACIO DE FORMAS.....	22
2.3.1 <i>Definición del espacio de formas</i>	22
2.3.1.1 <i>Espacio de similitudes Euclidianas</i>	23
2.3.1.2 <i>Espacio de formas planar afin</i>	24
2.3.2 <i>Normas en el espacio de formas</i>	25
CAPÍTULO 3: MODELO ESTÁTICO DETERMINÍSTICO	27
3.1 OBTENIENDO EL OBJETO A SEGUIR	27
3.1.1 <i>Encontrando discontinuidades mediante filtros</i>	29
3.1.2 <i>Substracción del fondo</i>	29
3.2 EL ALGORITMO DE AJUSTE.....	30
3.2.1 <i>Una formula simple</i>	31
3.2.1.1 <i>El regularizador</i>	31
3.2.2 <i>Algoritmo recursivo para ajuste</i>	33
3.2.3 <i>El algoritmo recursivo de ajuste: pseudo código</i>	35
CAPÍTULO 4: MODELO ESTÁTICO PROBABILÍSTICO	38
4.1 FÓRMULA DE BAYES.....	38
4.2 DISTRIBUCIÓN DE TRANSFORMACIONES: EL ALGORITMO DE AJUSTE	39
4.3 EL TÉRMINO REGULARIZADOR $P_0(X)$	40
4.3.1 <i>Variabilidad de un punto</i>	40
4.3.1.1 <i>Desplazamiento cuadrático medio en cada punto</i>	41
4.3.1.2 <i>Desplazamiento cuadrático medio promedio de una curva</i>	41
4.3.1.3 <i>Variabilidad de un punto sobre los segmentos normales</i>	41
4.3.1.4 <i>Variabilidad de un punto en la práctica</i>	42
4.3.2 <i>Puerta de validación: Selección de muestras</i>	46
CAPÍTULO 5: MODELO DINÁMICO PROBABILÍSTICO	50
5.1 EL MODELO PROBABILÍSTICO APLICADO AL SEGUIMIENTO	51
5.2 FUNCIONES DE DENSIDAD A PRIORI DINÁMICAS	52
5.2.1 <i>Función de densidad a priori global</i>	52
5.2.2 <i>Proceso de Markov Gaussiano de primer orden</i>	53
5.3 MOMENTOS AUTO-REGRESIVOS DE PRIMER ORDEN	55

5.4 MOMENTOS AUTO-REGRESIVOS DE SEGUNDO ORDEN	58
5.5 PARÁMETROS DINÁMICOS DEL MODELO	59
5.6 EL ESTADO DE EQUILIBRIO	61
5.7 APRENDIZAJE AUTOMÁTICO DE LOS PARÁMETROS	62
5.7.1 <i>Aprendizaje de los parámetros dinámicos en una dimensión</i>	64
5.7.2 <i>Aprendizaje de los parámetros dinámicos en el espacio de formas</i>	66
5.8 EL FILTRO DE KALMAN	67
5.8.1 <i>Predicción</i>	68
5.8.2 <i>Medición</i>	69
5.8.3 <i>Asimilación</i>	70
CAPÍTULO 6: IMPLEMENTACIÓN Y RESULTADOS	72
6.1 EL SISTEMA IMPLEMENTADO EN EL CONTEXTO DE LOS SISTEMAS DE SEGUIMIENTO	72
6.2 IMPLEMENTACIÓN DEL ALGORITMO DE AJUSTE	77
6.2.1 <i>Muestreo por longitud de arco</i>	77
6.2.2 <i>La curva característica: una entidad teórica</i>	78
6.2.3 <i>Optimización de la substracción del fondo</i>	78
6.3 IMPLEMENTACIÓN DEL FILTRO DE KALMAN	79
6.3.1 <i>Recuperación en presencia de incertidumbre</i>	79
6.3.2 <i>Simulando un estado de equilibrio</i>	81
6.4 TRABAJO CON VARIAS CURVAS	82
6.5 USANDO EL SISTEMA DE SEGUIMIENTO	84
6.5.1 <i>Algunos ejemplos</i>	84
6.5.2 <i>Ventajas del Aprendizaje</i>	88
6.5.3 <i>Evolución de la Incertidumbre</i>	95
6.5.4 <i>Desempeño del Sistema</i>	99
6.5.5 <i>Problemas encontrados</i>	105
6.6 MEJORAS POSIBLES Y TRABAJO A FUTURO	107
6.6.1 <i>Otros filtros para muestreo con segmentos normales</i>	107
6.6.2 <i>Muestreo con segmentos normales: eliminando el cuello de botella</i>	108
6.6.3 <i>Mejorando el modelo: El trabajo con gran cantidad de curvas</i>	109
6.6.4 <i>Multiprocesamiento y Procesamiento en Paralelo</i>	109
6.6.5 <i>Múltiples curvas patrón</i>	109
6.6.6 <i>Encontrando el sector inicial</i>	111
6.6.7 <i>Substracción del fondo sin cámara fija</i>	111
6.6.8 <i>Un sistema inteligente de seguimiento</i>	112
6.6.9 <i>Análisis de Componentes Principales (PCA)</i>	112
6.7 CONCLUSIONES FINALES	113
CAPÍTULO 7: APÉNDICE A	115
7.1 <i>¿POR QUÉ FUNCIONA EL ALGORITMO DE MINIMIZACIÓN PROPUESTO?</i>	115
7.2 <i>DEMOSTRACIÓN DEL CRECIMIENTO ILIMITADO DE LA VARIANZA</i>	118
7.3 <i>INVERTIBILIDAD DE LA MATRIZ B</i>	119
7.4 <i>CONDICIONES SOBRE β PARA ALCANZAR EL ESTADO DE EQUILIBRIO</i>	120
7.5 <i>DEMOSTRACIÓN DEL ORIGEN DEL ALGORITMO DE APRENDIZAJE AUTOMÁTICO DE PARÁMETROS</i>	123
CAPÍTULO 8: REFERENCIAS	127



Capítulo 1: Introducción

1.1 Objetivos del presente trabajo

El objetivo del presente trabajo es confeccionar un sistema sin *markers*¹ para realizar el seguimiento del cuerpo humano mediante *contornos activos*² que funcionan sobre el espacio de las transformaciones planares afines. Se hará hincapié en las optimizaciones necesarias para adecuar el sistema al trabajo en entornos de tiempo real, así como también para evitar el uso de hardware costoso y de difícil programación.

que es costoso y difícil.

Se hará un estudio de las restricciones inherentes del modelo de contornos activos que más afectan al desempeño del sistema y se propondrán soluciones a algunos de los problemas que se presentan.

Finalmente y con la idea de adecuar el sistema al seguimiento del cuerpo humano, se harán modificaciones al método estándar de seguimiento con contornos activos, para que soporte el funcionamiento con varias curvas en simultáneo.

1.2 Qué se entiende por seguimiento

Seguir un objeto es, básicamente, identificar su posición y acompañar sus movimientos a lo largo del tiempo mediante una curva que lo contornea. En el marco del presente trabajo, se utilizan secuencias de video que contienen el movimiento del objeto a seguir.

1.3 Trabajo previo realizado

Existen varios modelos para realizar seguimientos, los cuales varían mucho según el tipo de aplicación en la cual serán empleados. Entre los requerimientos más comunes de los sistemas de seguimiento se encuentran el funcionamiento en tiempo real, la detección de cierto tipo de movimiento, el trabajo con determinados objetos, etc. Los modelos propuestos para realizar seguimientos pueden agruparse de la siguiente forma [24]:

- *Basados en Movimiento*: Estos sistemas de seguimiento, se apoyan en métodos robustos para reconocer consistencias de movimiento en una secuencia de video. Estos métodos intentan reconocer el movimiento posible que pudo haber ocurrido entre un cuadro de video y el siguiente, mediante la comparación del objeto a

¹ Un *marker* es algún elemento distintivo que se aplica sobre el cuerpo con el objeto de facilitar su reconocimiento por medio de algoritmos específicos. Existen desde *markers* sofisticados que emplean electrodos, hasta otros más sencillos como pintura o ropa de colores contrastantes.

² Los *contornos activos* son un esquema en el cual una curva o contorno se deforma para adaptarse al contorno de algún objeto que se desee seguir.

seguir en ambos cuadros. Estos métodos permiten el reconocimiento tanto de objetos en 2D como en 3D. Su principal ventaja es que son rápidos (poco costosos computacionalmente). Sin embargo la desventaja que presentan es que su reconocimiento no tiene una semántica en el método: los métodos no saben si lo que reconocen es razonable o posible, y no asumen una representación o descripción del objeto que se está siguiendo.

- *Basados en Modelos*: Estos sistemas de seguimiento tienen subyacentemente una representación de alto nivel del objeto que desean seguir. *Saben* qué forma aproximada tiene y *conocen* las restricciones que puede presentar el mismo. Esta representación subyacente de un objeto puede ser en 2D o en 3D y su principal función es ayudar al sistema de seguimiento a filtrar los resultados *posibles* del algoritmo. Como es de esperar, son más costosos computacionalmente debido a que el método tiene que evaluar la factibilidad de los resultados dentro del modelo. Un caso muy claro de un modelo restringido se puede ver en [31] donde se aplica un modelo de Inverse Kinematics *Inversa cinemática*

Sin importar qué modelo utilice el sistema (incluso podría emplear un modelo híbrido), el seguimiento es realizado empleando mediciones hechas sobre la base de propiedades geométricas o basadas en regiones. De esto se desprenden las siguientes dos opciones para obtener las propiedades a medir de un objeto en una imagen:

- *Métodos Basados en Bordes*: Estos métodos se apoyan en la información provista por el contorno del objeto. Esta información no depende del tipo de movimiento ni tampoco de la forma del objeto, lo cual hace que puedan ser aplicados en cualquier tipo de secuencia de video. Usualmente los modelos basados en bordes emplean modelos de contornos activos tales como las *Snakes* o los *Contornos Activos Geodésicos*. Sin embargo, estos métodos requieren de una buena inicialización, ya que a partir de ésta se converge iterativamente hacia la solución final. Además, los métodos basados en bordes pueden sufrir de inestabilidad debido a que dependen demasiado de la información localmente obtenida, la cual puede ser imprecisa (puede ocurrir una oclusión, por ejemplo, con lo cual el método poco puede hacer).
- *Métodos Basados en Regiones*: Los métodos basados en regiones se apoyan en la información provista por una región de la imagen. Estos métodos suelen tener problemas cuando tienen que seguir el contorno de un objeto deformable, ya que la región de trabajo va cambiando constantemente. Por el contrario, son robustos debido a que emplean la información provista por toda una región (no miran sólo localmente como los métodos basados en bordes) y no dependen fuertemente de la inicialización. Ejemplos de estos métodos son los *Contornos Activos por Variables Fotométricas* [15], donde se emplean variables de visualización tales como el color para diferenciar el objeto a seguir.

Una vez elegido el método con el cual se obtendrán las propiedades a observar, se debe definir un método para comparar u operar con las mismas. Entre los más comunes figuran los siguientes métodos:

- *Minimización de Energía*: La idea de estos métodos, es la utilización de funciones de energía para acomodar una curva spline a la forma del objeto a seguir. Por ejemplo, si se tiene una observación del borde del objeto (obtenido por algún

método basado en bordes), el método basado en minimización de energía, tendrá disponible una función de energía que permitirá aproximar una curva hacia un borde. Cuando se emplea este, un contorno activo se describe por una función de energía que tiene dos componentes: la energía interna y la energía externa. La energía interna trata con las propiedades intrínsecas del contorno empleado para el seguimiento y establece restricciones para que los puntos que lo componen tengan una unidad. La energía interna se define como una suma entre una energía elástica (la cual permite al contorno contraerse y expandirse) y una energía de suavizado (la cual permite que el contorno sea suave o recto). La energía externa guía al contorno hacia el elemento de la imagen, definido por la propiedad observada (por ejemplo un borde). Su objetivo es llevar al contorno cerca del mínimo local deseado. A su vez también existe la energía de la imagen, la cual puede estar expresada como varios tipos de energía; por ejemplo la energía de línea es empleada para atraer a los contornos hacia líneas oscuras o claras dentro de la imagen. La energía total de un contorno activo, se expresa como la suma de su energía interna más su energía externa y la energía de la imagen. La función del contorno activo es minimizar la suma resultante para alcanzar el objetivo deseado, que es que el contorno se adapte a la forma y posición del objeto a seguir. Entre los sistemas de seguimiento basados en minimización de energía se pueden encontrar las *Snakes* [17] y los *Contornos Activos con Curva Patrón Deformable* [5] entre otros.

- *Minimización Geométrica*: Estos métodos se basan en consideraciones geométricas para hacer que el contorno se adapte al objeto que se desea seguir. Por ejemplo, supóngase que se ha calculado la distancia normal entre un punto de una spline deformable y un punto del contorno del objeto a seguir. Un método basado en minimización geométrica calcularía cual es la transformación que se debe aplicar al punto de control de la spline para que se mueva hasta la posición del punto del contorno del objeto. En [11] se puede ver un ejemplo de un algoritmo de *Snake* modificado³ para funcionar con un método basado en minimización geométrica.

Si bien estos métodos y modelos son opuestos es muy común el uso de sistemas de seguimiento que combinen distintos métodos y modelos, de forma de obtener lo mejor de unos y de otros [22][24] .

Algunos de los modelos más representativos en los cuales se basan los sistemas de seguimiento son:

- *Snakes*: La *snake* [17] se define como una spline que minimiza energía; la energía de la *snake* depende de su forma y de su ubicación dentro de la imagen. La spline que conforma la *snake* es atraída mediante fuerzas (determinadas por las funciones de energía⁴), hacia el contorno del objeto. Con la ayuda de un modelo físico e iterando hasta que la curva alcanza un equilibrio, este proceso llega a un punto en el cual se asume que la spline se ha acercado lo más posible

³ Típicamente los algoritmos de *Snake* funcionan con métodos de minimización de energía.

⁴ Tal como se mencionó anteriormente, esas funciones de energía pueden ser internas, externas o de la imagen.

al objeto. En el caso de las cavidades, el desempeño de la *snake* no es tan bueno como se desearía, ya que puede ocurrir que la spline quede atrapada en un mínimo local de la función de energía. Existen métodos que permiten salvar estos inconvenientes, mezclando curvas deformables (las cuales se pueden deformar, por ejemplo, en la dirección de una recta normal). Para más información ver [11]. Este modelo en general no soporta cambios en la topología del contorno mientras se va adaptando; esto quiere decir que la topología de la spline final obtenida es similar a la de la spline inicial (a menos que se implementen heurísticas o algún método alternativo). Esto resulta ser un problema cuando un número desconocido de objetos debe ser identificado, ya que la energía depende de la parametrización de las curvas, la cual no se halla relacionada con la geometría de los objetos.

- *Contornos Geodésicos*: El método de Contornos Geodésicos fue introducido como una alternativa geométrica para el método de las *Snakes* (las cuales emplean minimización de energía). Esto hace que los contornos geodésicos, a diferencia de los algoritmos clásicos de *Snake* y *Contornos Activos con Curva Patrón Deformable*, no se vean alterados por las posibles reparametrizaciones que puede sufrir una curva. Además, este método emplea un modelo muy simple y no precisa gran cantidad de información previa acerca del objeto. Sin embargo, en el caso de objetos rígidos, la *Snake* clásica se comporta de una manera más robusta. Aunque estos modelos no empleen minimización de energía para el movimiento del contorno hacia el objeto a seguir, pueden emplear funciones de energía equivalentes a hallar la curva geodésica que mejor interpreta al objeto a seguir⁵. Para más detalles ver [23].
- *Regiones Geodésicas*: Son una extensión de los *Contornos Activos Geodésicos*, pero, a diferencia de los *Contornos Geodésicos* que tenían información basada en bordes, incorporan información basada en regiones [24][25][26]. Su objetivo, es encontrar una partición (determinada por la curva más suave y de menor longitud que tenga en cuenta las propiedades del objeto a seguir), donde tanto la región interior como la región exterior preserven las propiedades deseadas de la imagen.
- *Contornos Activos por Variables Fotométricas*: Este método se basa en consideraciones fotométricas para aproximar una spline al objeto deseado. A diferencia de los métodos de variable geométrica (tales como detección de bordes), una variable fotométrica mide una cantidad de intensidad, de color o textura. Los objetos que pueden ser seguidos por este tipo de sistema de seguimiento se caracterizan por una función de probabilidad en alguna variable fotométrica (por ejemplo, se identifican con un color característico). En cada cuadro, el algoritmo intenta buscar en qué región se genera la distribución sobre la variable fotométrica en cuestión que más se asemeja a la distribución del modelo. Un sistema de seguimiento que funciona de esta manera, no necesita confiar en los bordes, evitándose problemas tales como bordes espúreos, bordes que no se pueden detectar por el contraste difuso, etc. Aún en el caso de un detector de bordes ideal, los seguidores del método de *Contornos Activos por Variables*

⁵ En algunos casos particulares y dadas ciertas condiciones, las *Snakes* han probado ser equivalentes a un subconjunto de contornos geodésicos. Para más detalles, referirse a [12].

Fotométricas opinan que los métodos que funcionan por detección de bordes nunca podrán ser totalmente efectivos debido a que no toman en cuenta la rica información presente en las variables fotométricas. Los seguidores del método afirman que, por ejemplo, se puede ahorrar mucho tiempo en un sistema que siga el movimiento de los labios, si se considera que los labios humanos no tienen muchos colores distintos. Los *Contornos Activos por Variables Fotométricas* [15] también han resultado efectivos funcionando en videos que tengan fondos difusos o con mucho ruido. Como contrapartida, este tipo de método no es apto para el funcionamiento en tiempo real, debido a su alto costo computacional.

1.4 Qué problemas puede presentar el seguimiento

Los problemas que se presentan a la hora de seguir un objeto son numerosos, y sus fuentes muy variadas.

Entre los más comunes se encuentran (ver Figura 1-1):

- Seguimiento en tiempo real: Las películas actuales pueden tener aproximadamente 24 cuadros por segundo. Es sumamente costoso realizar todos los cálculos requeridos para modificar la curva que representa al objeto en esa pequeña fracción de tiempo (asumiendo que no se cuenta con un supercomputador).
- Oclusión: A lo largo de la película es posible que el objeto a seguir sea ocultado por otros objetos presentes en la escena.
- Deformaciones: El objeto puede cambiar su forma y posición en los sucesivos cuadros de la película.
- Ruido: Puede existir distorsión que impida distinguir claramente el objeto en una escena.

A lo largo del presente trabajo, se plantearán distintos modelos (con sus respectivas restricciones) que dan una respuesta al problema del seguimiento. Además se explicaran distintas técnicas que solucionan los problemas presentados en el seguimiento.

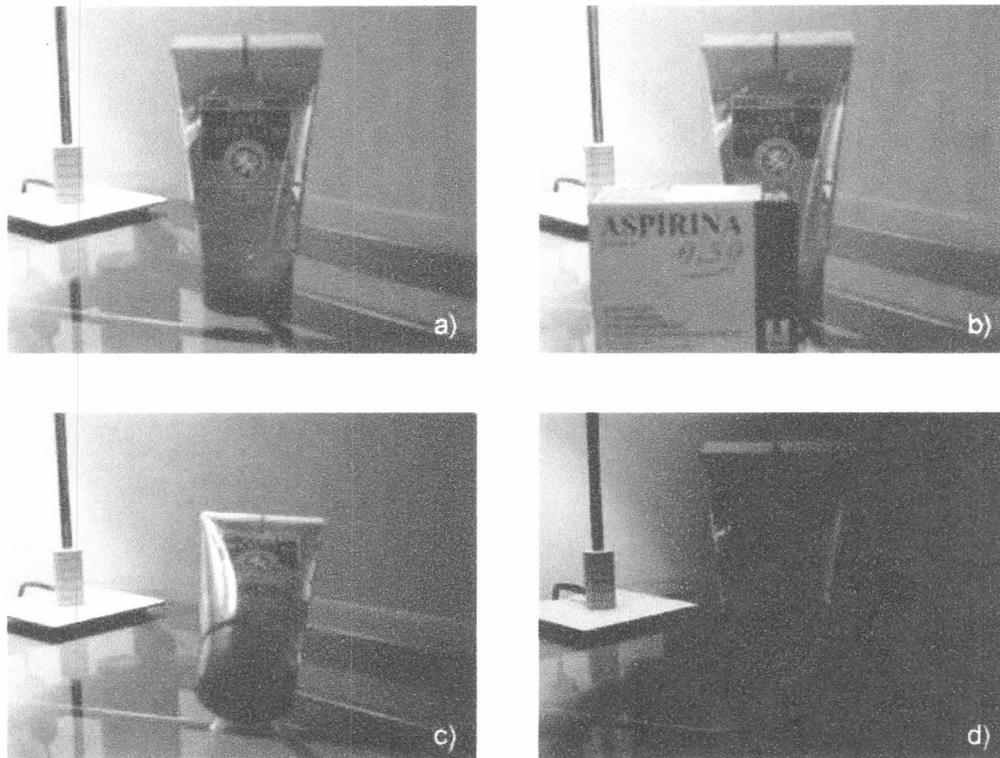


Figura 1-1. En la figura se pueden ver algunos de los problemas que se presentan a la hora de realizar el reconocimiento de un objeto dentro de una escena (lo cual se extiende naturalmente al seguimiento de un objeto en una película). En **a)** el objeto puede ser reconocido sin problemas en la escena. En **b)** existe otro objeto que está ocluyendo al primero: esto puede hacer que el algoritmo de ajuste empleado para el reconocimiento de objetos (ver más adelante en la sección 3.2) crea que el objeto tiene una forma distinta a la real. En **c)** el mismo objeto ha sufrido un cambio en su forma, producto de una rotación en el eje paralelo al plano de la imagen. Finalmente en **d)** se ha introducido ruido en forma de penumbra, lo cual dificulta el reconocimiento del objeto.

1.5 Qué se verá en los siguientes capítulos

En el capítulo 2 se hará una revisión al tema de representaciones de curvas y demás estructuras de datos que serán empleadas en el transcurso del presente trabajo. Es importante comprender bien el comportamiento de las estructuras de datos para saber de qué manera reaccionan ante ciertos estímulos, qué se puede lograr que hagan y cuáles son sus restricciones.

En el capítulo 3 se introducirá un *modelo estático determinístico*. Este modelo se denomina así debido a que opera sobre una única imagen y no hay cálculos estadísticos involucrados. Este modelo es fundamental para el funcionamiento del sistema de seguimiento de videos debido a que un video es una secuencia de imágenes estáticas: un sistema que realiza un seguimiento en su más simple expresión podría ser un conjunto de llamadas a un sistema que emplea un modelo estático determinístico. El modelo estático determinístico propuesto establece también el método por el cual se inspeccionan las imágenes y, como se verá más adelante, es un factor clave a la hora de optimizar el

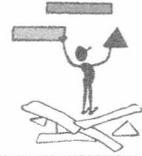
código para su funcionamiento en tiempo real. Finalmente, y quizás lo más importante, es que este modelo determina una representación (en forma de *transformación*) del objeto que fue inspeccionado en la imagen de forma tal que pueda ser empleado en procesamientos posteriores. Esa representación tiene incidencia, entre otras cosas, en cuestiones tales como velocidad de funcionamiento en tiempo real, precisión del análisis de la imagen y restricciones generales sobre los tipos de objetos que se pueden reconocer en una imagen.

En el capítulo 4 se verán los mismos conceptos que fueron tratados en el capítulo 3, pero desde el punto de vista probabilístico. Es decir que se planteará un *modelo estático probabilístico*. El modelo de seguimiento en secuencias de video tiene una base estadística muy fuerte sobre la que debe apoyarse y parte de esa base se plantea en este capítulo. Si bien a la hora de realizar los análisis sobre una imagen estática se emplean los conceptos del capítulo 3, es importante mostrar que existe un marco de trabajo estadístico que le da sustento a los modelos dinámicos. Por otro lado, se hacen algunos agregados al método del capítulo anterior para adaptarlo a su posterior uso en el seguimiento de video.

En el capítulo 5 se introduce por primera vez el *modelo dinámico*. En este modelo el objetivo es seguir un objeto en una secuencia de video. Esto involucra una gran cantidad de elementos tanto de procesamiento estático de imágenes (capítulo 3) como de probabilidad y estadística. Se emplearán aquí modelos que facilitan la predicción de comportamiento tales como la cadena de Markov y los momentos auto-regresivos. Se harán también análisis y comparaciones con respecto al alcance o no por parte del sistema de un estado de equilibrio, de acuerdo al modelo que se haya empleado. También se introducirá en este capítulo un mecanismo que permite el aprendizaje de los parámetros dinámicos del sistema de seguimiento. Finalmente se presentará el algoritmo que será el encargado de dirigir todo el proceso de seguimiento de un objeto en una secuencia de video: el filtro de Kalman.

En el capítulo 6 se analiza la implementación realizada en el presente trabajo. Se comentarán las mejoras introducidas a los algoritmos y métodos conocidos, y se comentarán las ventajas y desventajas que presenta el modelo elegido con respecto a los otros modelos existentes. Asimismo se comentarán algunas de las cuestiones de implementación que se consideran de mayor relevancia y se mostrarán ejemplos de corrida y gráficos comparativos de los distintos métodos. Por último se planteará qué temas han quedado abiertos a partir del trabajo realizado y qué cosas se podrían desarrollar en el futuro con el objeto de mejorar el sistema presentado.

Capítulo 2: Splines y espacio de formas



En el marco del presente trabajo, todas las curvas intervinientes serán representadas mediante curvas *splines* paramétricas. Estas curvas son representaciones computacionales de la forma $(x(s), y(s))$ donde s es un parámetro que se incrementa a medida que se recorre la curva y x e y son funciones de s conocidas como splines. Una spline de orden d es una función polinomial que consiste en *segmentos* polinomiales de orden d concatenados.

Las curvas paramétricas son atractivas porque permiten representar eficientemente los bordes de las distintas formas presentes en una imagen (Figura 2-1). Ciertos objetos simples pueden ser representados utilizando unos pocos segmentos, mientras que otros objetos más complicados se pueden representar elevando el orden polinomial d , aunque, en la práctica es preferible aumentar el número de segmentos utilizados.

Generalmente se utilizan los órdenes polinomiales cuadráticos ($d=3$) y cúbicos ($d=4$). Utilizar polinomios de bajo grado, aún ante la presencia de objetos complejos, permite realizar cálculos simples y estables.



Figura 2-1. Esta imagen muestra cómo se representan los bordes de los objetos mediante curvas splines paramétricas.

2.1 Funciones B-spline

Las funciones B-splines son una representación computacional conveniente de las funciones splines. En la forma de B-spline, la función spline $x(s)$ se construye como la suma ponderada de N_B funciones base⁶ $B_n(s)$, con $n = 0, \dots, N_B - 1$. En el caso más simple ("regular"), cada función base contiene d polinomios cada uno definido sobre un segmento. Estos segmentos se unen en puntos denominados *nodos*.

Más adelante, en la sección 2.1.1, se mostrará cómo calcular las funciones bases que intervienen en una función B-spline.

La función spline construida es (ver Figura 2-2):

$$x(s) = \sum_{n=0}^{N_B-1} x_n B_n(s) \quad (2-1)$$

donde x_n son los pesos que se aplican a las funciones bases $B_n(s)$ respectivas. Esta suma puede ser expresada, en la notación matricial, como

$$x(s) = B(s)^T Q^x \quad (2-2)$$

donde,

$$B(s) = (B_0(s), B_1(s), \dots, B_{N_B-1}(s))^T \quad \text{y} \quad Q^x = \begin{pmatrix} x_0 \\ \vdots \\ x_{N_B-1} \end{pmatrix} \quad (2-3)$$

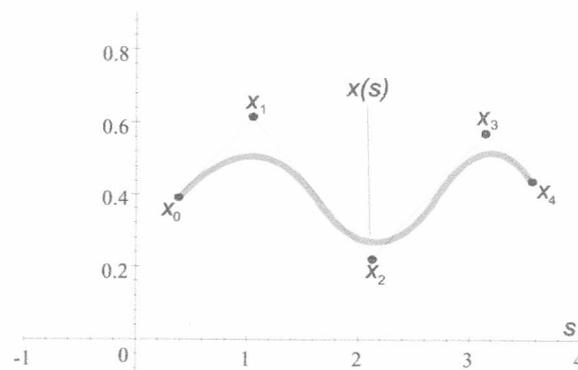


Figura 2-2. En esta figura se pueden observar los pesos x_0, x_1, x_2, x_3, x_4 aplicados a las distintas bases para obtener una función spline.

⁶ De ahí el nombre de 'B'-splines.

2.1.1 Funciones bases

Las funciones spline se escriben como una combinación lineal de un número de *funciones bases*. Estas últimas se construyen utilizando la regla general que se muestra a continuación según sea la distribución de los nodos (uniones de los segmentos).

Funciones B-Spline uniformes: En este caso, los nodos no se repiten y se encuentran espaciados en intervalos unitarios.

Caso Base:

$$B_{n,1}(s) = \begin{cases} 1 & \text{si } n \leq s < n+1 \\ 0 & \text{si } \text{no} \end{cases}$$

Paso iterativo:

$$B_{n,d}(s) = \frac{(s-n)B_{n,d-1}(s) + (n+d-s)B_{n+1,d-1}(s)}{d-1}$$

Estas funciones satisfacen las siguientes propiedades

Soporte: $B_{n,d}(s) = 0$ para todo $s \notin [n, n+d)$.

Positividad: $B_{n,d}(s) \geq 0$ para todo s .

Normalización: $\sum_{-\infty}^{\infty} B_{n,d}(s) = 1$ (para todo s).

Invarianza traslacional: $B_{n+1,d}(s) = B_{n,d}(s-1)$ para todo s .

Ejemplo: En Figura 2-3 se muestra la primer función base de una B-spline cuadrática con nodos espaciados uniformemente en intervalos unitarios.

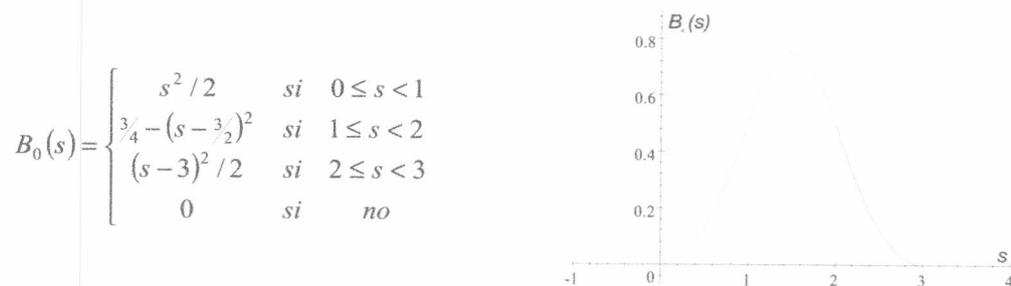


Figura 2-3. B-Spline cuadrática ($d=2$) con nodos en $s = 0, 1, 2, 3, 4$. Se considera cada segmento de longitud unitaria.

Funciones B-Spline NO uniformes: En este caso, los nodos se encuentran espaciados arbitrariamente. Considérese una spline con N_k nodos en los valores $k_0 \leq k_1 \leq \dots \leq k_{N_k-1}$.

Caso Base:

$$B_{n,1}(s) = \begin{cases} 1 & \text{si } k_n \leq s < k_{n+1} \\ 0 & \text{si } \text{no} \end{cases}$$

Paso iterativo:

$$B_{n,d}(s) = \frac{(s - k_n)B_{n,d-1}(s)}{k_{n+d-1} - k_n} + \frac{(k_{n+d} - s)B_{n+1,d-1}(s)}{k_{n+d} - k_{n+1}}$$

Las funciones bases no uniformes, se reducen al caso uniforme si se considera $k_n = n$.

2.1.2 Bases finitas

Cuando $s \rightarrow \pm\infty$, la cantidad de bases utilizadas (B_n) en una spline es infinita. Computacionalmente, es necesario contar con un número finito de bases al momento de considerar aplicaciones prácticas.

Existen dos tipos de bases finitas de una spline. Estas pueden ser periódicas (Figura 2-4) ó aperiódicas (Figura 2-5) sobre un intervalo $0 \leq s \leq L$.

En el caso periódico, se consideran las funciones bases de forma cíclica. Las bases son B_0, \dots, B_{L-1} ($N_B = L$).

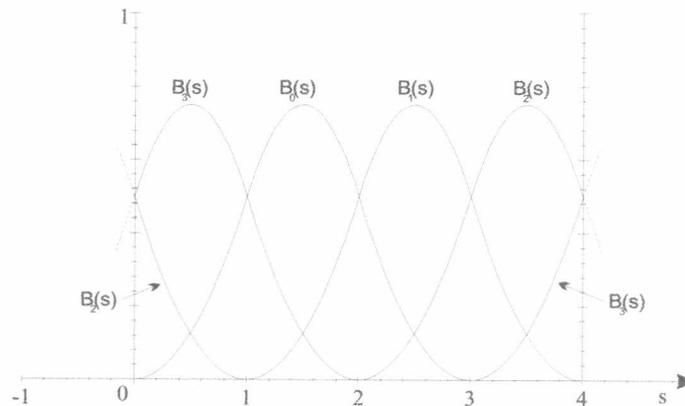


Figura 2-4. En esta figura puede verse cómo cada función base es una copia trasladada de la anterior pero teniendo en cuenta el efecto cíclico donde la periodicidad lo demanda.

En el caso aperiódico, se requieren de los llamados *nodos múltiples* en los extremos. No sigue siendo válido que el número de funciones bases N_B es igual a la longitud del intervalo L . Trabajar con nodos múltiples permite obtener curvas que contengan picos y discontinuidades.

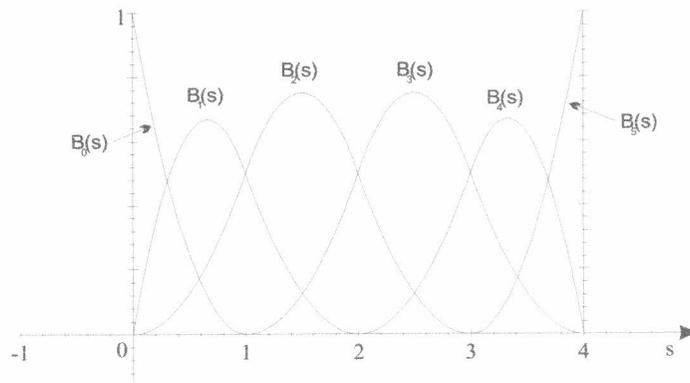


Figura 2-5. Estas funciones bases no son una traslación de la anterior como ocurría en el caso periódico. Se requieren de nodos de multiplicidad d en los extremos.

Existe un algoritmo eficiente para generar funciones spline a partir de los pesos x_n , en el cual, las funciones bases se representan como una matriz de coeficientes polinomiales y las mismas se encuentran precalculadas. Esto permite contar con un ahorro sustancial de tiempo a la hora de evaluar una spline. Para más detalles referirse a [5].

2.1.3 Norma y producto interno de funciones spline

Es muy útil calcular la *norma* L_2 ($\|x\|$) de una función $x(s)$

$$\|x\|^2 = \frac{1}{L} \int_0^L x(s)^2 ds \quad (2-4)$$

Esta norma representa el valor *raíz-cuadrático-medio* y es especialmente útil para medir la diferencia entre dos funciones $x_1(s), x_2(s)$ como $\|x_1 - x_2\|$.

La norma tiene un correspondiente *producto interno* cuya sintaxis es $\langle \cdot, \cdot \rangle$ y se aplica a un par de funciones x, y como $\langle x, y \rangle$.

La relación existente entre el producto interno y la norma es $\langle x, x \rangle = \|x\|^2$, por lo tanto el producto interno entre dos funciones queda determinado como

$$\langle x, y \rangle = \frac{1}{L} \int_0^L x(s)y(s)ds \quad (2-5)$$

Debido a que se están representando las funciones de forma compacta, utilizando vectores de pesos de las splines, se desea expresar las normas y los productos internos en término de dichos vectores, esto es, definir $\|\cdot\|$ para los vectores de pesos de forma tal que

$$\|Q^x\| = \|x\| \quad (2-6)$$

En base a (2-2)

$$\|x\|^2 = (Q^x)^T \left(\frac{1}{L} \int_0^L B(s)B(s)^T ds \right) Q^x \quad (2-7)$$

De esta forma, la norma debe definirse como

$$\|Q^x\| = \sqrt{(Q^x)^T B Q^x} \quad (2-8)$$

donde la *matriz de métrica* (B) de una función B-spline se define como

$$B = \frac{1}{L} \left(\int_0^L B(s)B(s)^T ds \right) \quad (2-9)$$

El producto interno se define como

$$\langle Q_1^x, Q_2^x \rangle = (Q_1^x)^T B Q_2^x \quad (2-10)$$

Las matrices de métricas tienen la propiedad de ser esparzas. Esto se debe al hecho que cada peso Q_n^x afecta a la función $x(s)$ solo en un pequeño subintervalo (el soporte de la correspondiente función base B_n) y esto permite obtener algoritmos de cuadrados mínimos eficientes.

Aproximación de funciones

Utilizando el producto interno, ciertos problemas de aproximación se resuelven de manera simple. Por ejemplo, se puede utilizar el producto interno para aproximar una función spline representada como Q^x en término de otras dos funciones spline Q_1^x y Q_2^x . La aproximación de *cuadrados mínimos* se expresa, utilizando el producto interno, como

$$\hat{Q}^x = \begin{pmatrix} Q_1^x & Q_2^x \end{pmatrix} \begin{pmatrix} \langle Q_1^x, Q_1^x \rangle & \langle Q_1^x, Q_2^x \rangle \\ \langle Q_2^x, Q_1^x \rangle & \langle Q_2^x, Q_2^x \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle Q_1^x, Q^x \rangle \\ \langle Q_2^x, Q^x \rangle \end{pmatrix}$$

Otro tipo importante de problemas es el de aproximar una función $f(s)$, que no necesariamente es una spline, como una función spline $x(s)$, representada por sus pesos Q^x . Nuevamente, la solución puede obtenerse utilizando el producto interno. Por ejemplo

$$Q^x = B^{-1} \frac{1}{L} \int B(s)f(s)ds$$

La aproximación de funciones se utiliza para construir aproximaciones de curvas presentes en las imágenes, es decir, obtener una spline que represente a dicha curva. Naturalmente, no es posible evaluar integrales en las imágenes, por lo tanto, los datos se deben muestrear. La aproximación como una spline de una función discretizada es

$$Q^x = B^{-1} \frac{1}{N} \sum_{n=1}^N B(s_n) f(s_n)$$

2.2 Curvas paramétricas B-Spline

Las curvas spline paramétricas tienen coordenadas $x(s), y(s)$ cada una de las cuales es una función spline del parámetro de la curva s .

$$r(s) = (x(s), y(s)) \quad (2-11)$$

Es necesario elegir un intervalo apropiado $0 \leq s \leq L$ para cubrir L segmentos y también un conjunto de funciones bases $B_0, B_1, \dots, B_{N_B-1}$. Si el intervalo $[0, L]$ se considera periódico, la curva paramétrica resultante será cerrada.

Para cada función base B_n , se debe definir un *punto de control* $q_n = (q_n^x, q_n^y)^T$. La curva se obtiene como el resultado de la suma ponderada de estos puntos de control

$$r(s) = \sum_{n=0}^{N_B-1} B_n(s) q_n \quad \text{para } 0 \leq s \leq L \quad (2-12)$$

2.2.1 Vector de control

Trabajar explícitamente con los puntos de control hace engorrosa la notación. Es por ello que se buscará una notación más compacta.

Sea S_Q el espacio de vectores de control Q que contienen las coordenadas de los puntos de control, primero todas las x -coordenadas y luego todas las y -coordenadas.

$$Q = \begin{pmatrix} Q^x \\ Q^y \end{pmatrix} \quad \text{donde } Q^x = \begin{pmatrix} q_0^x \\ \vdots \\ q_{N_B-1}^x \end{pmatrix} \quad (2-13)$$

y análogamente para Q^y . Luego, las funciones de las coordenadas se pueden escribir como

$$x(s) = B(s)^T Q^x \quad y(s) = B(s)^T Q^y \quad (2-14)$$

donde $B(s)$ es un vector de bases B-spline definido en (2-3).

Luego,

$$r(s) = U(s)Q \quad \text{para } 0 \leq s \leq L \quad (2-15)$$

donde

$$U(s) = \begin{pmatrix} B(s)^T & 0 \\ 0 & B(s)^T \end{pmatrix} \quad (2-16)$$

es una matriz de $2 \times N_Q$.

2.2.2 Norma y producto interno de curvas

Una vez obtenida una representación de las curvas spline paramétricas, se deberá extender la definición de la norma y el producto interno para las curvas. Se definirá la norma $\|\cdot\|$ para las curvas B-spline como

$$\|Q\|^2 = \frac{1}{L} \int_{s=0}^L |r(s)|^2 ds \quad (2-17)$$

ó, equivalentemente, por (2-15) y (2-9)

$$\|Q\|^2 = Q^T U Q \quad (2-18)$$

donde la *matriz de métrica* para curvas (U) se define en términos de la matriz de métrica (B) para funciones B-spline.

$$U = \frac{1}{L} \int_0^L U(s)^T U(s) ds \quad (2-19)$$

ó, equivalentemente,

$$U = \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix} \quad (2-20)$$

La norma también determina un producto interno para curvas

$$\langle Q_1, Q_2 \rangle = Q_1^T U Q_2 \quad (2-21)$$

Comparación entre curvas

La norma de las curvas es particularmente útil cuando se utiliza como medida de comparación entre dos curvas utilizando la distancia $\|Q_1 - Q_2\|$.

Existen otras normas más simples que la norma definida en (2-18), por ejemplo, la norma Euclideana $|Q|^2 = Q^T Q$ del vector de control.

La norma Euclideana es mucho más simple de calcular ya que la matriz U se reemplazaría por la matriz de identidad. Igualmente, más allá de la simplificación

obtenida, la norma Euclídeana no funciona satisfactoriamente. El ejemplo de la Figura 2-6 muestra como la norma Euclídeana no es buena para medir la cercanía de las curvas.

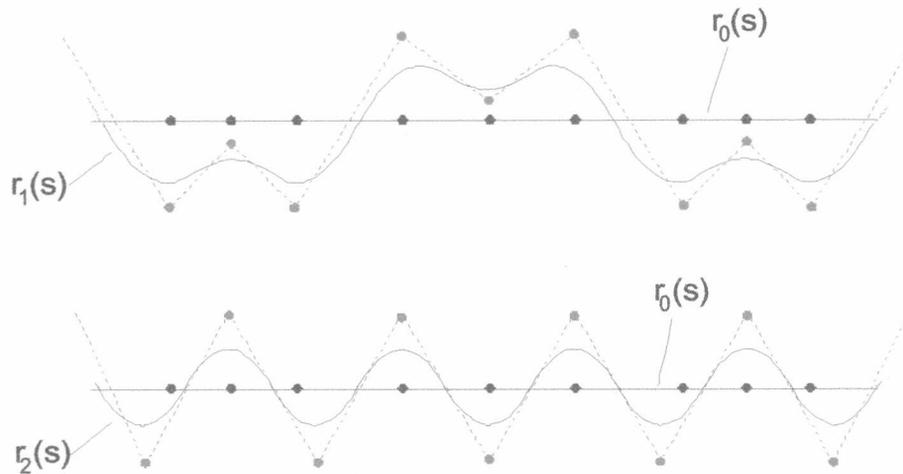


Figura 2-6. En el ejemplo se muestran las curvas r_0 , r_1 y r_2 . La curva r_2 está más cerca de la curva r_0 de lo que lo está la curva r_1 . A pesar de esto, considerando la distancia Euclídeana, se puede ver por la posición de los puntos de control que r_2 está más lejos de la curva r_0 de lo que lo está la curva r_1 . Cortesía de A. Blake y M. Isard.

2.2.3 Reparametrización de curvas

Es importante notar que la norma de curvas como magnitud de la diferencia entre dos curvas no acepta transparentemente la reparametrización de una de las mismas. Por ejemplo, una curva $r(s)$, $0 \leq s \leq L$ puede ser reparametrizada para obtener una nueva curva $r^*(s) = r(L-s)$, $0 \leq s \leq L$. Geométricamente las dos curvas son idénticas y se esperaría que la diferencia entre las curvas tenga una norma de cero, sin embargo esto no es lo que ocurre.

Una solución general al problema de la reparametrización requeriría una búsqueda sobre las posibles parametrizaciones. La proximidad de una curva $r(s)$ a una segunda curva $r^*(s)$ podría evaluarse como

$$\min_g \|r(s) - r^*(g(s))\| \quad (2-22)$$

donde el mínimo se explora sobre un espacio de funciones de reparametrización g , usando un procedimiento de optimización apropiado. Se han desarrollado poderosos procedimientos de optimización pero son computacionalmente costosos. Una opción más económica se obtiene al usar una medida de distancia $d(r, r^*)$ que es invariante a una menor-reparametrización de la curva r . La distancia se basa en el desplazamiento *normal* de r relativo a r^* , omitiendo cualquier componente tangencial de desplazamiento. En la sección 3.2.2, se desarrollará este tema.

2.3 Espacio de formas

En la práctica, es bastante deseable distinguir entre el *vector-spline* $Q \in S_Q$ que describe la forma básica de un objeto y el *vector de formas* $X \in S_X$, donde S_X es un *espacio de formas*. Mientras que S_Q es un espacio de vectores de B-splines y tiene dimensiones $N_Q = 2N_B$, el espacio de formas S_X se construye en base a un espacio de vectores de dimensión N_X , que es comúnmente menor que N_Q . El espacio de formas se define como todas las posibles transformaciones que pueden ser aplicadas a una curva.

La necesidad de hacer esta distinción se puede observar en la Figura 2-7. El manejo arbitrario del vector-spline Q es demasiado general para ser interesante en la práctica y lleva a trabajar con contornos activos que son inestables.

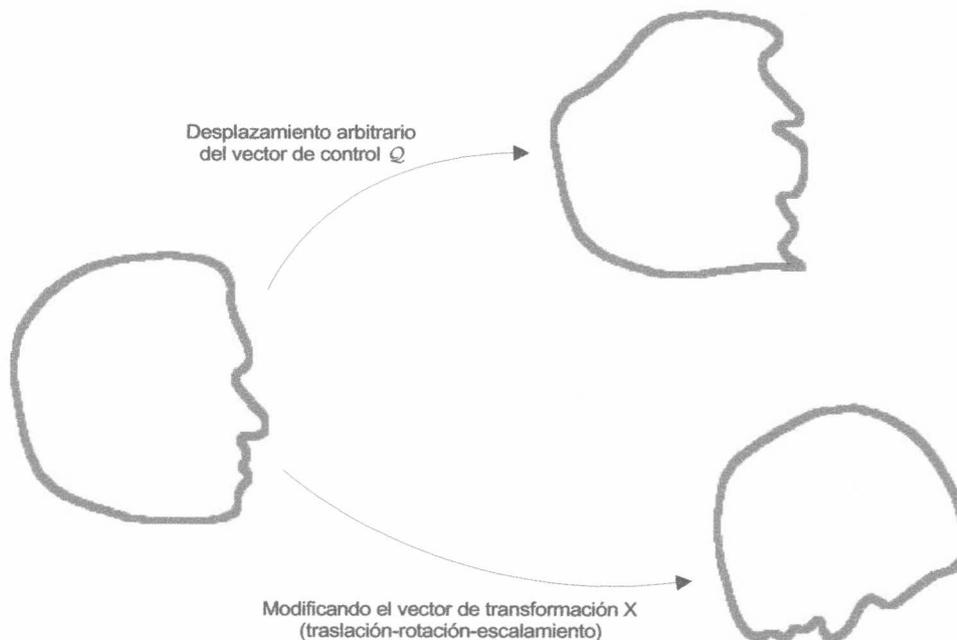


Figura 2-7. En esta figura se puede observar como la curva que representa una cabeza deja de lucir como tal cuando se manipulan los puntos de control arbitrariamente. Utilizando una clase restringida de transformaciones parametrizadas por un vector de configuraciones X se obtienen resultados más interesantes.

2.3.1 Definición del espacio de formas

Un espacio de formas $S_X = \mathcal{L}(W, Q_0)$ es un mapeo lineal de un vector del espacio de formas $X \in R^{N_X}$ a un vector de spline $Q \in R^{N_Q}$

$$Q = WX + Q_0 \quad (2-23)$$

donde W es la *matriz de formas* de $N_Q \times N_X$. La constante de desplazamiento Q_0 es una curva base contra la cual se miden las variaciones. Por ejemplo, el conjunto de formas representando a Q_0 y a otras curvas cercanas a Q_0 puede expresarse restringiendo el espacio de formas S_X a pequeños X .

2.3.1.1 Espacio de similitudes Euclidianas

Este espacio de formas de 4 dimensiones representa exactamente las variaciones que puede presentar un objeto desplazándose sobre un plano mientras es observado por una cámara fija con aumento (zoom) situada perpendicularmente a dicho plano. El efecto del objeto es que el mismo se mueve rígidamente en sus dimensiones y que el mismo rota sobre un eje perpendicular al plano en que se mueve. Estos movimientos poseen 4 *grados de libertad* que se pueden observar en la Figura 2-8.

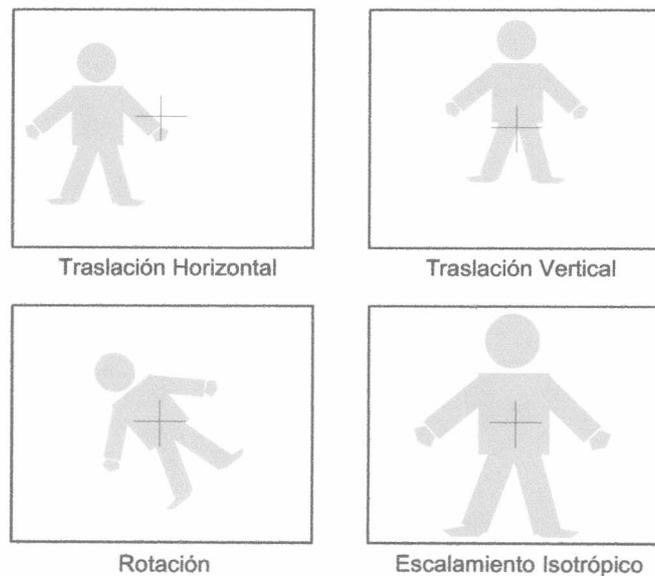


Figura 2-8. El espacio de formas de similitudes Euclidianas tiene 4 grados de libertad. En la figura se pueden observar aplicadas a un cuerpo.

Las similitudes Euclidianas de una curva $r_0(s)$ representada por Q_0 forman un espacio de formas de 4 dimensiones cuya matriz de formas está determinada por

$$W = \begin{pmatrix} \mathbf{1} & \mathbf{0} & Q_0^x & -Q_0^y \\ \mathbf{0} & \mathbf{1} & Q_0^y & Q_0^x \end{pmatrix} \quad (2-24)$$

donde los N_B -vectores $\mathbf{0}$ y $\mathbf{1}$ son

$$\mathbf{0} = (0,0,\dots,0)^T \quad \text{y} \quad \mathbf{1} = (1,1,\dots,1)^T$$

La primera y la segunda columna de W actúan sobre las traslaciones horizontales y verticales respectivamente mientras que la tercera y la cuarta cubren las rotaciones y el escalamiento.

A continuación se muestran algunos ejemplos de representaciones de formas en el espacio de similitudes Euclidianas

- $X = (0,0,0,0)^T$ representa la curva original Q_0 .
- $X = (1,0,0,0)^T$ representa la curva Q_0 trasladada una unidad a la derecha.
- $X = (0,0,1,0)^T$ representa la curva Q_0 aumentada al doble de su tamaño.
- $X = (0,0, \cos\theta - 1, \sin\theta)^T$ representa la curva Q_0 rotada según el ángulo θ .

2.3.1.2 Espacio de formas *planar afin*

Otro espacio de formas importante es el espacio *planar afin*. Este espacio posee 6 *grados de libertad*, los cuales alcanzan para describir gran parte de las transformaciones que puede sufrir un objeto ubicado paralelo al plano de la imagen. En la Figura 2-9 se puede observar el conjunto de transformaciones de este espacio de formas. Las primeras 3 transformaciones corresponden a las 3 primeras presentadas en el espacio de similitudes Euclidianas. Las 3 últimas transformaciones generan un subespacio que incluye la cuarta transformación del espacio de similitudes Euclidianas y agrega 2 grados de libertad para el escalamiento direccional. Este escalamiento se produce cuando un objeto planar rota sobre un eje que es paralelo al plano sobre el que se mueve.

Las transformaciones planares afin de una curva $r_0(s)$ (representada por Q_0) pueden ser descritas por un espacio de formas de 6 dimensiones cuya matriz de formas está determinada por

$$W = \begin{pmatrix} 1 & 0 & Q_0^x & 0 & 0 & Q_0^y \\ 0 & 1 & 0 & Q_0^y & Q_0^x & 0 \end{pmatrix} \quad (2-25)$$

Las primeras dos columnas de W representan las traslaciones horizontales y verticales respectivamente. Los restantes movimientos afines, los cuales no corresponden uno a uno con las últimas columnas de W , pueden expresarse como combinación lineal de dichas columnas. Dado que las transformaciones del espacio de formas están determinadas por la ecuación (2-23), se puede ver que los elementos de X actúan como pesos en las columnas de W .

Algunos ejemplos de transformaciones son los siguientes:

- $X = (0,0,0,0,0,0)^T$ representa la curva original Q_0 .
- $X = (1,0,0,0,0,0)^T$ representa la curva Q_0 trasladada una unidad a la derecha.
- $X = (0,0,1,1,0,0)^T$ representa la curva Q_0 aumentada al doble de su tamaño.

- $X = (0,0,\cos\theta - 1,\cos\theta - 1,-\sin\theta,\sin\theta)^T$ representa la curva Q_0 rotada según el ángulo θ .
- $X = (0,0,1,0,0,0)^T$ representa la curva Q_0 ensanchada al doble de su tamaño.

El espacio de formas planar afín, es el espacio de formas que se utilizará en la implementación que se presenta.

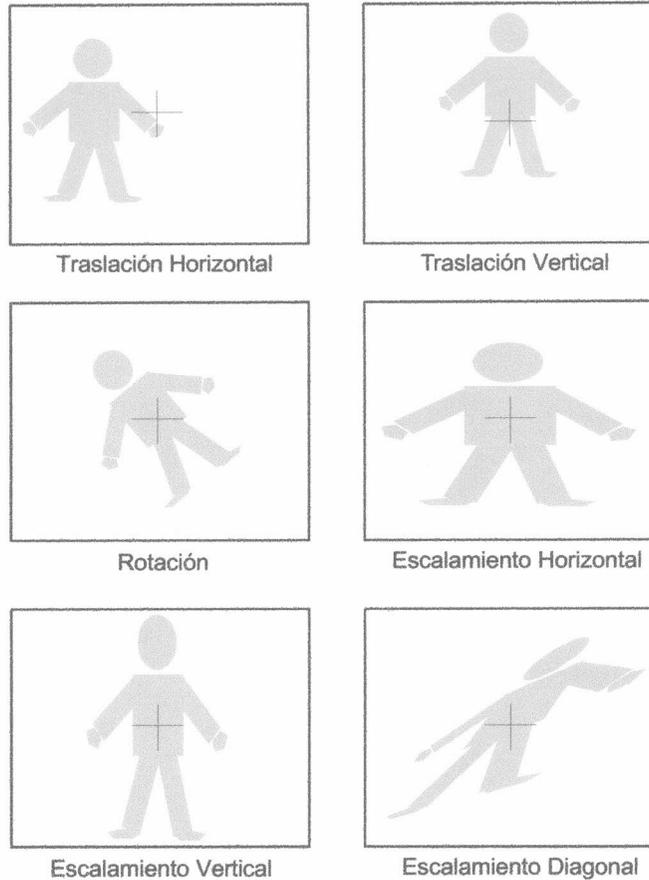


Figura 2-9. El conjunto de transformaciones planares afín posee 6 grados de libertad.

2.3.2 Normas en el espacio de formas

Dado que generalmente es preferible trabajar con el espacio de formas S_X , es necesario contar con una fórmula para calcular la norma del parámetro X . Se deberá tener una definición consistente de forma tal que, para un espacio dado, $\|Q_1 - Q_2\| = \|X_1 - X_2\|$. La norma L_2 en el espacio de formas S_X se dice que es *inducida* de la norma definida sobre S_Q , la cual fue inducida sobre la norma L_2 , definida a su vez sobre el espacio de curvas $r(s)$. Según (2-23), ésto se obtiene definiendo

$$\|X\| = \|\sqrt{X^T \mathcal{H} X}\| \quad (2-26)$$

donde

$$\mathcal{H} = W^T U W \quad (2-27)$$

La norma definida sobre S_X , tiene la siguiente interpretación geométrica

$$\|X\| = \|Q - Q_0\|$$

Esto significa que representa el desplazamiento promedio de la curva parametrizada por X con respecto a la curva base.

Además, se puede definir un mapeo entre S_Q y el espacio de formas S_X . En general no existe una inversa del mapeo W desde S_Q a S_X , pero si las columnas de W son linealmente independientes, se puede definir la pseudo-inversa W^+

$$X = W^+(Q - Q_0) \quad \text{donde } W^+ = \mathcal{H}^{-1} W^T U$$

Capítulo 3: Modelo estático determinístico



Para comprender el presente modelo, es necesario definir los siguientes términos:

- **Imagen:** Una imagen se define como una función $f: Z^2 \rightarrow [0,255] \times [0,255] \times [0,255]$. De esta forma, una imagen permite realizar una proyección entre una coordenada en el plano entero y un color de la escala RGB.
- **Curva patrón ($\bar{r}(s)$):** Esta curva modela el contorno de un objeto, imponiendo restricciones sobre los posibles objetos que se pudieran encontrar. Es a esta curva a la que se le aplicarán las deformaciones necesarias para imitar el movimiento del objeto contenido en la imagen. Esta curva se representa mediante una spline (Q_0).
- **Curva característica ($r_f(s)$):** Esta curva representa el contorno del objeto contenido en la imagen.

El objetivo del modelo estático determinístico es encontrar una transformación tal que, aplicada a la curva patrón, obtenga por resultado un contorno similar al objeto presente en la imagen. Dicha transformación X (donde X pertenece al espacio planar afín) satisface la siguiente condición

$$\min_X \|WX + Q_0 - Q_f\|^2$$

La solución de esta ecuación requiere de un algoritmo específico (algoritmo de ajuste) y un conjunto de restricciones que se aplican al mismo. Estas restricciones deben asegurar que la transformación obtenida mantenga una similitud con el objeto a seguir.

3.1 Obteniendo el objeto a seguir

El objeto a seguir se encuentra contenido en una escena, razón por la cual puede estar mezclado con otros objetos que no son de interés para el algoritmo y que pueden confundirlo. El algoritmo de ajuste debe saber en qué sector de la imagen se encuentra el objeto con el que trabajará.

Para simplificar el problema, en esta primera sección se asume que el algoritmo de ajuste sabe en qué sector de la imagen se encuentra el objeto a seguir. Con esta información, el algoritmo debe extraer la curva característica del objeto.

Dado que $\bar{r}(s)$ es una spline y está representada por puntos de control (Q_0), es necesario obtener una representación análoga del contorno del objeto contenido en la imagen. Esta representación es necesaria para poder comparar ambas curvas y es $r_f(s)$ quien representa el contorno del objeto contenido en la imagen.

Para encontrar la curva característica, se ubica la curva patrón en el sector de la imagen indicado. La idea es ir avanzando⁷ de a intervalos predefinidos sobre la curva patrón y en cada uno de ellos realizar un muestreo mediante segmentos normales a la curva con la finalidad de obtener los puntos que forman parte de la curva característica. La región determinada por la curva patrón y las líneas normales conforman lo que se denomina *sector de interés* (ver Figura 3-1).

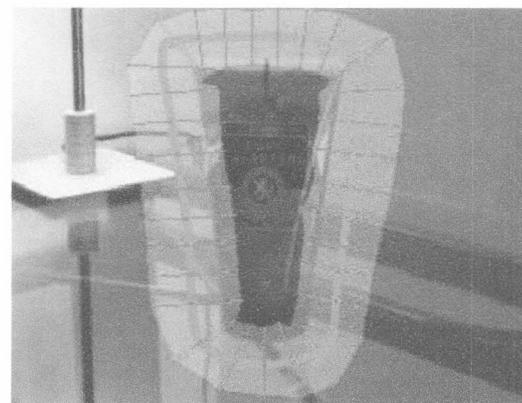
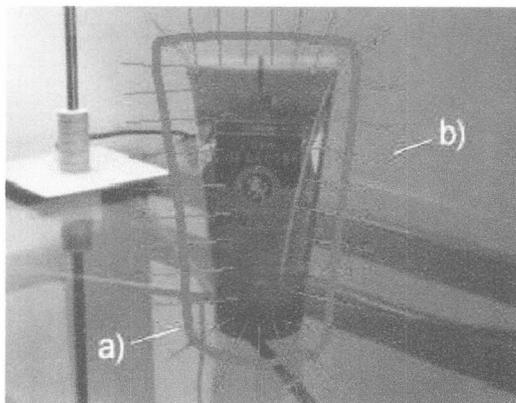


Figura 3-1. En la imagen del centro (arriba) se puede ver un objeto contenido en una escena. En la imagen de la izquierda (abajo) se observa en a) a la curva $\bar{r}(s)$ y en b) los segmentos normales trazados a partir de ésta. En la imagen de la derecha (abajo) se puede ver cómo esas mismas normales determinan un sector de interés dentro del cual se realiza la búsqueda del contorno del objeto.

⁷ La forma en la que se va avanzando por la curva, es en el sentido creciente del parámetro. Recuérdese que estas curvas tienen una representación paramétrica (Véase sección 2.2 Curvas paramétricas B-Spline).

La finalidad del muestreo de normales es obtener los puntos que forman parte de la curva característica. Para ello se recorre cada una de las normales a la curva, con el objetivo de reconocer discontinuidades en la intensidad de los píxeles atravesados, indicando así la ubicación del contorno del objeto. Las discontinuidades se pueden hallar empleando filtros.

3.1.1 Encontrando discontinuidades mediante filtros

Los segmentos normales sobre los que se muestrea, requieren de un filtro para reconocer dónde empieza el contorno del objeto que se está siguiendo. Estas zonas donde se reconoce que comienza el contorno del objeto, se denominan de *alto contraste*. La idea es aplicar un operador o filtro a la señal de intensidad muestreada de manera que en los puntos donde ésta supera cierto umbral preestablecido, se encuentren los cambios de contraste o intensidad buscados.

El muestreo de normales utilizando filtros para encontrar discontinuidades, posee una gran desventaja al trabajar con imágenes de alto contraste. Es posible encontrar discontinuidades que no son las esperadas. Por ejemplo, si el objeto que se desea encontrar está cerca de otro objeto⁸ cuya intensidad de color es reconocida por el método como una discontinuidad, es posible que el algoritmo produzca resultados erróneos. Por otro lado, si la imagen no tiene un buen contraste, encontrar discontinuidades será más difícil ya que el filtro no podrá distinguir ningún objeto.

Para solucionar estos inconvenientes se puede emplear un método conocido como *substracción del fondo*.

3.1.2 Substracción del fondo

Encontrar un objeto de color sobre un fondo negro, es una tarea sencilla. Sin embargo, éste no será el caso general, ya que como fue planteado anteriormente, el objeto se encuentra contenido en una escena. Una técnica muy utilizada para separar objetos del fondo de la escena, se conoce como *substracción del fondo*. Esta técnica tiene como objetivo descartar el fondo de una escena para prevenir que el mismo pueda llegar a distraer al método empleado para el reconocimiento, haciéndolo pensar erróneamente que se ha encontrado al objeto.

Para poder abstraer el fondo, se almacena una imagen del fondo de la escena $I_B(x,y)$ previa a la inclusión del objeto deseado. Dada una imagen que posee al objeto incluido en ella, la detección de borde se restringe a las áreas $I(x,y)$ que cumplen la siguiente condición:

$$|I(x,y) - I_B(x,y)| > \sigma$$

donde σ es un parámetro de tolerancia.

⁸ Ambos objetos se encuentran dentro del mismo sector de interés.

Puede surgir un problema si el color del objeto seguido coincide con el color del fondo en algún píxel, ya que dicho punto será eliminado por la substracción del fondo. Para evitar esto, se puede cambiar el valor del parámetro de tolerancia. Este parámetro define clases de equivalencia en las que se ubican los colores (mientras más alta la tolerancia, habrá menos clases de equivalencia y el sistema distinguirá una menor cantidad de colores como distintos)⁹.

Una vez substraído el fondo, se emplea un filtro sencillo para reconocer al objeto deseado.

La técnica de substracción del fondo presenta las siguientes restricciones:

- El color del objeto puede coincidir con el color del fondo de la escena en la misma posición. Esto hará que se *substraiga* parte del objeto.
- Es necesario contar con un fondo fijo, en el cual el objeto no se halle incluido. Además se requiere que la cámara se encuentre estática al momento de realizar el seguimiento.
- Aplicar esta técnica a toda la imagen es computacionalmente muy costoso.



Figura 3-2. En esta figura se puede ver un ejemplo del funcionamiento del método de substracción de fondo. En la primera imagen el objeto se encuentra contenido en la escena. La segunda imagen, que fue previamente almacenada, es substraída de la primera. Como resultado se obtiene una tercera imagen, en la que se puede observar que el fondo ha sido substraído. También puede observarse que el objeto no se ha mantenido intacto.

3.2 El algoritmo de ajuste

El algoritmo de ajuste empleado en el presente trabajo, como se mencionó anteriormente, tiene por objeto encontrar una transformación tal que, aplicada a $\bar{r}(s)$ obtenga por resultado un contorno *similar* a $r_f(s)$.

⁹ Dada la forma en la que funciona una cámara, si se toman dos imágenes de la misma escena en tiempos distintos (aunque sean cortos), éstas nunca serán exactamente iguales. Esto ocurre debido a que existen constantes variaciones en la luz, que impactan de manera distinta en el CCD de la cámara. Por este motivo, es necesario el uso del valor de tolerancia, el cual se incluye para minimizar esas variaciones de luz (haciendo que en tiempos cortos haya una menor diferencia entre colores similares).

3.2.1 Una formula simple

En su expresión más simple, el algoritmo de ajuste se puede expresar con la siguiente fórmula:

$$\min_X \|WX + Q_0 - Q_f\|^2 \quad (3-1)$$

Esta expresión busca minimizar la diferencia entre la curva patrón transformada ($WX + Q_0$), y la curva real observada (Q_f). Hallar el valor de este X no es complicado, y se puede aproximar como

$$\hat{X} = W^+(Q_f - Q_0) \quad (3-2)$$

Donde W^+ es la pseudo-inversa de la matriz del espacio de formas que se está empleando. Analíticamente $W^+ = \mathcal{H}^{-1}W^T U$.

3.2.1.1 El regularizador

El resultado obtenido en la ecuación (3-2) se puede mejorar aún más, si se agrega un término regularizador. Este término es empleado para indicar cuánto se desea que la transformación obtenida aproxime a la curva patrón. La curva resultante es solución de

$$\min_{r(s)} \alpha \|r - \bar{r}\|^2 + \|r - r_f\|^2 \quad (3-3)$$

En esta última fórmula se pueden observar dos términos bien definidos: El primero de ellos ($\alpha \|r - \bar{r}\|^2$) contempla el regularizador, mientras que el segundo ($\|r - r_f\|^2$) da una métrica de la distancia entre curvas.

El problema se puede expresar más convenientemente como

$$\min_X \alpha \|X - \bar{X}\|^2 + \|Q - Q_f\|^2 \quad \text{donde } Q = WX + Q_0 \quad (3-4)$$

El mínimo X que soluciona la ecuación (3-4), satisface simultáneamente lo siguiente:

- Regularización: Es la transformación que hace mínima su diferencia con \bar{X} (la transformación que da lugar a la curva patrón original).
- Medida de desplazamiento de curvas: Minimiza la diferencia entre el objeto encontrado (Q_f) y la curva patrón transformada ($Q = WX + Q_0$).

La transformación X obtenida determina una nueva curva Q , la cual está influenciada por la curva patrón. En la práctica puede ser deseable que Q_0 inflencie solamente en la forma de Q y no así en su posición u orientación. Para lograr ésto se requiere de un regularizador más general. La fórmula (3-4) se reescribe como

$$\min_X (X - \bar{X})^T S (X - \bar{X}) + \|Q - Q_f\|^2 \quad \text{donde } Q = WX + Q_0 \quad (3-5)$$

El regularizador $((X - \bar{X})^T \bar{S} (X - \bar{X}))^{10}$ en este caso) es un término que establece el grado de similitud entre una curva y la curva patrón. Existe un factor, llamado α , contenido en la matriz \bar{S} , que indica cuánta influencia de la curva patrón recibirá la transformación resultado de la ecuación (3-5). En la Figura 3-3 se puede ver un ejemplo del uso de este factor. Puede existir un valor de regularizador distinto para cada uno de los grados de libertad que tenga el espacio de formas que se esté empleando¹¹. La matriz \bar{S} actúa como un filtro y los grados de libertad que se verán afectados por el regularizador, son considerados en su construcción.

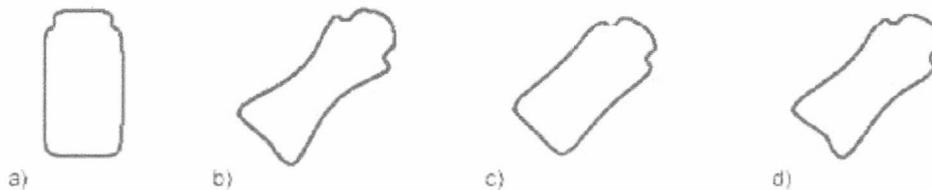


Figura 3-3. En a) se puede ver la curva patrón original. En b) se puede ver a la curva patrón transformada para ajustar a un objeto determinado. En c) se observa la misma transformación solo que esta vez se ha aplicado un regularizador, con factor $\alpha = 3$. En d) se puede ver nuevamente la misma transformación, pero con un regularizador con factor $\alpha = 0.5$. Se puede ver que mientras mayor sea el valor de α , más se parecerá el resultado a forma de la curva patrón.

El operador de proyección

El operador de proyección sirve para restringir transformaciones a un subespacio invariante o a su complemento según sea necesario.

Para obtener la invarianza deseada del regularizador sobre un subespacio $\mathcal{S}_s \subset \mathcal{S}$ de transformaciones, por ejemplo el de las similitudes Euclidianas, se debe restringir a \bar{S} mediante un operador de proyección E^d para que trabaje sobre las deformaciones fuera del subespacio invariante \mathcal{S}_s .

$$\bar{S} = \alpha E^{dT} \mathcal{H} E^d$$

El operador de proyección se puede expresar en términos de la matriz de formas W_s para el subespacio y su matriz de formas pseudo inversa W_s^+

¹⁰ Este término no será empleado para el seguimiento dinámico, ya que el método empleado en este trabajo no lo utiliza.

¹¹ Para obtener un regularizador invariante sobre el subespacio de transformaciones (posición y orientación), \bar{S} debe estar restringida a un conjunto de operaciones que contemplen sólo deformaciones.

$$E^d = I - E^s \quad \text{donde } E^s = W^+ W_s W_s^+ W$$

3.2.2 Algoritmo recursivo para ajuste

Existen dos tipos de desplazamiento posibles que puede sufrir una curva: el tangencial y el normal. El desplazamiento tangencial es aproximadamente el que se da avanzando por la curva, sin salir de la misma; representa una reparametrización de las curvas. El desplazamiento normal es la distancia entre dos curvas.

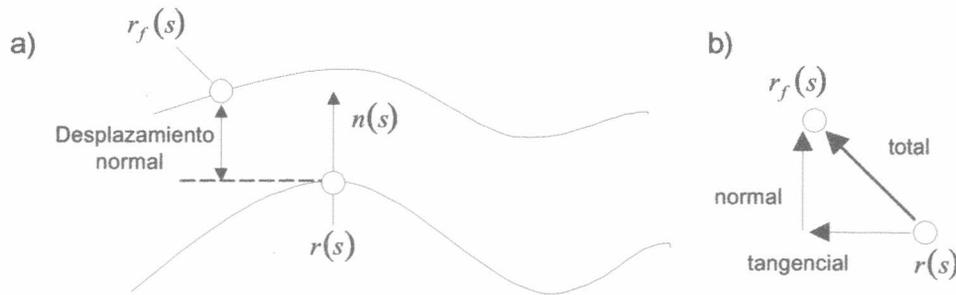


Figura 3-4. En a) se observa el desplazamiento normal entre dos curvas. En b) se muestra en detalle la descomposición de los desplazamientos entre dos puntos. Cortesía de A. Blake-M. Isard

Para el funcionamiento correcto del algoritmo de ajuste, se deben tener en cuenta las siguientes restricciones:

- El algoritmo no considera los desplazamientos tangenciales. Por esta razón, es una restricción que la diferencia de parametrización entre ambas curvas no sea *grande*. Siendo g una función de parametrización, se asume que $g(s) \approx s$, donde s es el parámetro de las splines.
- El desplazamiento normal no debe ser demasiado grande.

De estas restricciones se concluye que el algoritmo realiza pequeños ajustes. No respetarlas hará que se obtengan resultados erróneos (ver Figura 3-5).

Dadas las limitaciones enunciadas, para medir el desplazamiento entre curvas se utilizará en (3-3) la fórmula (3-6) en lugar de $\|r - r_f\|^2$. Para una derivación más completa referirse a [5].

$$d(r, r_f) \approx \frac{1}{L} \int [(r(s) - r_f(s)) \cdot n(s)]^2 ds \quad (3-6)$$

Se define ahora una norma $\|\cdot\|_{\bar{n}}$, con respecto a una curva patrón $\bar{r}(s)$, cuyas normales son $\bar{n}(s)$

$$\|r\|_{\bar{n}}^2 \equiv \frac{1}{L} \int [r(s) \cdot \bar{n}(s)]^2 ds \quad (3-7)$$

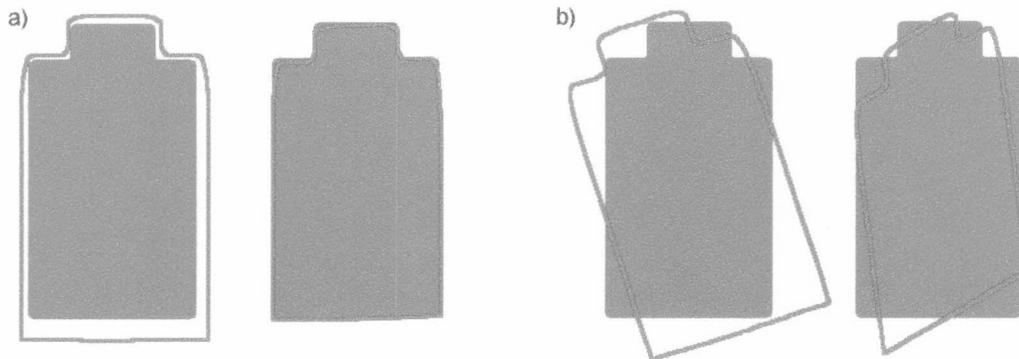


Figura 3-5. En a) y b) se puede observar un objeto y un sector de interés antes y después del algoritmo de ajuste. En a) el sector de interés cumple con las restricciones del algoritmo de ajuste, por lo que el ajuste se realiza correctamente. En b) las restricciones no se cumplen: el sector considerado tiene un desplazamiento tangencial importante respecto de la curva característica que *contornea* al objeto, por lo cual el ajuste no se realiza correctamente.

Esta norma tiene la propiedad de aproximar a (3-6) cuando es aplicada a la diferencia entre curvas. Esto último ocurre si las curvas que intervienen en la medición se hallan suficientemente cerca de $\bar{r}(s)$. De esta forma

$$\|r - r_f\|_{\bar{n}} \approx d(r, r_f) \quad (3-8)$$

La fórmula (3-6) puede ser llevada al caso discreto, mediante una sumatoria, dando como resultado

$$\|r - r_f\|_{\bar{n}}^2 \approx \frac{1}{N} \sum_{i=1}^N [(r_f(s_i) - r(s_i)) \cdot \bar{n}(s_i)]^2 \quad (3-9)$$

Dado que $r(s_i)$ puede ser expresada explícitamente en función de la transformación X como $r(s_i) = U(s_i)(WX + Q_0)$, se puede expresar la norma de la diferencia como

$$\|r - r_f\|_{\bar{n}}^2 \approx \frac{1}{N} \sum_{i=1}^N (v_i - h(s_i))^T [X - \bar{X}]^2 \quad (3-10)$$

donde $v_i = [r_f(s_i) - \bar{r}(s_i)] \cdot \bar{n}(s_i)$ se denomina *innovación*, y $h(s_i)^T = \bar{n}(s_i)^T U(s_i)W$. Asimismo, y tal como fuera mostrado en 2.1.3, se tiene que

$$\|Q - Q_f\|^2 \equiv \|r - r_f\|^2 \approx d^2(r, r_f) \quad (3-11)$$

Reemplazando en (3-5), se obtiene

$$\min_X (X - \bar{X})^T \bar{S} (X - \bar{X}) + \frac{1}{N} \sum_{i=1}^N (v_i - h(s_i))^T [X - \bar{X}]^2 \quad (3-12)$$

El algoritmo que se implementa, también contempla la ponderación de ciertos sectores de la curva más que otros. Puede ser posible que se le quiera dar un peso mayor a algún sector de la curva, por ejemplo a zonas donde se requiere un mayor detalle. Modificando (3-12) para incorporar este mecanismo, se obtiene la siguiente ecuación que se debe minimizar

$$\min_X (X - \bar{X})^T \bar{S} (X - \bar{X}) + \sum_{i=1}^N \frac{1}{\sigma_i^2} (v_i - h(s_i))^T [X - \bar{X}]^2 \quad (3-13)$$

Si se consideran todos los puntos con el mismo peso, o sea $\sigma_i = \sqrt{N}$, se obtiene la ecuación (3-12).

3.2.3 El algoritmo recursivo de ajuste: pseudo código

En esta sección se mostrará un pseudo código¹² del algoritmo de ajuste empleado para la implementación del modelo estático determinístico. Se mostrarán y explicarán brevemente los puntos más significativos.

Se quiere hallar $\min_X (X - \bar{X})^T \bar{S} (X - \bar{X}) + \sum_{i=1}^N \frac{1}{\sigma_i^2} (v_i - h(s_i))^T [X - \bar{X}]^2$:

1. Se eligen muestras s_i , donde $i = 1 \dots N$ y $s_1 = 0, s_{i+1} = s_i + h, s_N = L$.
2. Para cada i aplicar un filtro a lo largo de una línea normal a $\bar{r}(s_i)$, tratando de encontrar los puntos del objeto $r_f(s_i)$.
3. Inicializa $r Z_0 = 0, S_0 = 0$
4. Para $i = 1 \dots N$
 - 4.1. $v_i = (r_f(s_i) - \bar{r}(s_i)) \cdot \bar{n}(s_i)$
 - 4.2. $h(s_i)^T = \bar{n}(s_i)^T U(s_i) W$
 - 4.3. $S_i = S_{i-1} + \frac{1}{\sigma_i^2} h(s_i) h(s_i)^T$
 $Z_i = Z_{i-1} + \frac{1}{\sigma_i^2} h(s_i) v_i$
5. El vector de observación es $Z = Z_N$ y la matriz de información estadística es $S = S_N$
6. Finalmente, la curva que mejor ajusta a la curva patrón, se puede hallar mediante la transformación $\hat{X} = \bar{X} + (\bar{S} + S)^{-1} Z$.

Figura 3-6. Algoritmo recursivo de ajuste.

¹² La implementación del algoritmo en el lenguaje de programación elegido, puede tener diferencias con el código aquí mostrado. Estas diferencias obedecen principalmente a cuestiones de optimización. Con el fin de simplificar la comprensión, se ha hecho la adaptación que aquí se detalla.

1. Lo primero que hace el algoritmo es seleccionar los puntos de la curva patrón, en los cuales se muestrearán los segmentos normales. Estos puntos son unívocamente determinados por el valor del parámetro de la curva (s en este caso), el cual se incrementa una cantidad h para obtener cada uno de los valores.
2. En cada uno de los puntos seleccionados, se traza un segmento normal a la curva. En muchos de los casos, el segmento normal encontrará en su camino al contorno del objeto a reconocer. La intersección entre el segmento normal y el contorno deseado da por resultado un punto $r_f(s_i)$, el cual representa la muestra tomada. Estos puntos son almacenados para su posterior procesamiento.
3. Se inicializan los valores de S y Z . Inicialmente no hay información estadística, por lo cual $S = 0$. Asimismo, dado que todavía no se han hecho observaciones, el valor de Z es igual a 0.
4. Una vez obtenidos tanto los valores de la curva patrón ($\bar{r}(s_i)$) como los valores de la curva característica ($r_f(s_i)$), se puede comenzar la comparación de los mismos.
 - 4.1. En este punto del algoritmo, se obtiene la distancia entre ambas curvas en los puntos $r_f(s_i)$ y $\bar{r}(s_i)$. Esta distancia se toma en dirección al segmento normal a la curva $\bar{r}(s)$, en el punto $\bar{r}(s_i)$ (ver Figura 3-4.a). Esta distancia, representada por v_i , se denomina *innovación*. Este valor es útil para la detección y posterior eliminación de *outliers*¹³, ya que puede detectarse erróneamente un punto de otro objeto presente en la escena. La diferencia en la posición entre este punto erróneamente detectado y $\bar{r}(s_i)$ puede ser importante; eso se evidencia en el valor de $|v_i|$. De esta forma, se puede descartar la muestra (en la sección 4.3.2, se mostrará una modificación a este algoritmo para eliminar *outliers*).
 - 4.2. En este punto se realiza un almacenamiento temporal en el vector $h(s_i)$. Este vector es una estructura auxiliar, la cual carece de semántica en el contexto del algoritmo de ajuste.
 - 4.3. En este punto se observa cómo se va confeccionando la matriz de estadística S y el vector de observación Z . El término σ^2 es denominado error de medición; en su cálculo se incluye el error en el punto considerado o lo que más adelante se verá como *incertidumbre*. Cuando el valor de σ^2 es grande para un determinado $r(s_i)$, se evidencia una mayor incertidumbre en ese punto. Por consiguiente $r(s_i)$ tendrá un menor peso en la observación Z y en la matriz S . La matriz S indica la incertidumbre global que se tiene en el algoritmo; si sus componentes se mantienen pequeñas, la incertidumbre es mayor. Todo esto será visto en detalle cuando se explique el algoritmo empleado para el seguimiento: *el filtro de Kalman*.

¹³ Un outlier surge cuando la muestra encontrada no pertenece al objeto a seguir, ya sea porque el mismo es ocultado parcialmente por algún otro objeto de la escena o porque otro objeto se ha reconocido.

5. El *vector de observación* es el n -ésimo valor de Z . La *matriz de información estadística* es el n -ésimo valor de S .
6. Finalmente se devuelve la transformación resultado del algoritmo, aquí denotada como \hat{X} .

Para una demostración formal del funcionamiento del algoritmo de minimización, refiérase al Apéndice A-¿Por qué funciona el algoritmo de minimización?



Capítulo 4: Modelo estático probabilístico

En esta sección se introduce un modelo probabilístico que opera sobre imágenes estáticas y utiliza al algoritmo de ajuste presentado anteriormente. El concepto de ajustar una curva por regularización se reinterpreta en esta sección de forma probabilística.

Se presenta un modelo teórico, y se muestra su equivalencia con el modelo estático del capítulo anterior. Se introducen conceptos que permiten manejar y medir la incertidumbre en las observaciones. También se presenta un método para detección y posterior eliminación de *outliers*. Todo esto permite sentar las bases para los modelos dinámicos, ya que estos últimos se apoyan fuertemente en conceptos probabilísticos.

Para comprender mejor el modelo estático probabilístico, se definirá el término *incertidumbre* como:

La incertidumbre en un sistema de seguimiento se puede definir como la capacidad del modelo de predecir el comportamiento del objeto en condiciones donde no hay observación. Si existe una baja incertidumbre en el sistema, éste sabe dónde se encuentra y cuál es la forma que tiene el objeto a seguir sin necesidad de realizar observaciones.

4.1 Fórmula de Bayes

A diferencia del modelo presentado en el capítulo anterior, el modelo que aquí se presenta, no obtiene como resultado una única transformación sino que permite obtener una familia de transformaciones posibles, expresada como una función de densidad gaussiana

$$p(X | r_f) \quad (4-1)$$

Esta función representa la probabilidad de una determinada transformación, dado que se ha observado cierto dato r_f . El valor donde alcanza su máximo (denominado \hat{X} , y el cual coincide con la media de la distribución), es la transformación que se está buscando.

De acuerdo a la fórmula de Bayes, la función de densidad (4-1) puede expresarse como el producto entre una *densidad a priori* $p_0(X)$ y una *densidad de observación* $p(r_f | X)$:

$$p(X | r_f) \propto p(r_f | X)p_0(X) \quad (4-2)$$

Donde ambos factores son funciones de densidad gaussianas.

Si se maximiza en ambos lados de (4-2) queda

$$\hat{X} = \max_X p(X | r_f) \propto \max_X (p(r_f | X)p_0(X)) \quad (4-3)$$

En este punto se pueden observar las similitudes existentes entre los modelos probabilístico y determinístico.

- El término $p_0(X)$ de la ecuación (4-3) es equivalente al término $\alpha \|r - \bar{r}\|^2$ de la ecuación (3-3), puesto que ambos funcionan como regularizador. El valor que se quiere por resultado en ambas expresiones es \bar{X} : en el primer caso se busca maximizar la probabilidad de una distribución gaussiana cuya media es \bar{X} ¹⁴, mientras que en el segundo caso se busca minimizar una distancia con \bar{X} .
- El término $p(r_f | X)$ de la ecuación (4-3) es equivalente al término $\|r - r_f\|^2$ de la ecuación (3-3), debido a que ambos representan la observación realizada por el algoritmo de ajuste (Véase la sección 3.2). El valor buscado en ambos casos es r_f , ya que es el valor que maximiza $p(r_f | X)$ y que minimiza $\|r - r_f\|^2$.

4.2 Distribución de transformaciones: el algoritmo de ajuste

Tal como se menciona al comienzo del capítulo, el algoritmo de ajuste interpretado probabilísticamente, permite obtener como resultado el máximo de un conjunto de transformaciones. Estas transformaciones siguen una distribución gaussiana $\mathcal{N}(\bar{X}, S^{-1})$, cuya función de densidad asociada es $p(r_f | X)$ (de acuerdo a la ecuación (4-2)).

La *matriz de covarianza* (S^{-1}) de la distribución seguida por las transformaciones obtenidas con el algoritmo de ajuste, nos da una medida de qué tan confiable es el resultado del mismo. Si los valores de las componentes de la matriz S^{-1} son grandes, esto indicará una mayor incertidumbre, o lo que es lo mismo, la transformación devuelta por el algoritmo de ajuste no es tan confiable. Más adelante se verá cómo este mecanismo es empleado por el *filtro de Kalman* para darle un mayor o menor peso a la observación de acuerdo a la incertidumbre presente en el sistema.

El resultado obtenido por el algoritmo de ajuste tanto para el caso determinístico como para el caso probabilístico es el mismo: En el caso determinístico, se busca minimizar $\|r - r_f\|^2$, mediante una transformación X tal que $Q = WX + Q_0$ (donde Q representa al vector de control de r y Q_0 representa al vector de control de \bar{r}). Idealmente, si $Q \approx Q_f$ (con lo cual $r \approx r_f$), se hace mínima la diferencia entre ambas curvas. En el caso probabilístico se busca maximizar $p(r_f | X)$; si se da una transformación X tal que $Q_f \approx WX + Q_0$, entonces $p(r_f | X) \approx 1$.

¹⁴ Esto se verá en las siguientes secciones.

4.3 El término regularizador $p_0(X)$

Tal como fuera mencionado anteriormente, el problema de la regularización consiste en hallar el mínimo de $(X - \bar{X})^T \bar{S} (X - \bar{X})$. En (4-2) el término $p_0(X)$ es empleado como regularizador y representa una función de densidad gaussiana cuya media es \bar{X} . Este término se puede expresar como

$$p_0(X) = \frac{1}{(\sqrt{2\pi})^{N_x}} \frac{1}{\sqrt{\det P_0}} \exp -\frac{1}{2} (X - \bar{X})^T \bar{S} (X - \bar{X}) \quad (4-4)$$

El problema presente en la regularización consiste en hallar el máximo de esta función de densidad cuya distribución es $\mathcal{N}(\bar{X}, P_0)$.

Se puede observar que $p_0(X)$ en (4-4) decrece cuando X se va alejando de \bar{X} . La inversa de \bar{S} (si es que existe) es la matriz de covarianza P_0 , una medida de la variabilidad que puede sufrir una curva con respecto a la distribución a la que pertenece.

4.3.1 Variabilidad de un punto

La matriz de covarianza P_0 también puede ser empleada para calcular $P_r(s)$, una matriz que indica la variabilidad posicional de un punto dado $r(s)$ sobre una curva. Esto indica que, si se considera cada punto de la curva como una variable aleatoria $r(s) \forall s$, la distribución de los mismos será $\mathcal{N}(\bar{r}(s), P_r(s))$.

Por otro lado, calculando la variación en un punto y usando propiedades se tiene que la matriz de covarianza viene dada por:

$$\begin{aligned} P_r(s) &= \mathcal{E}[(r(s) - \bar{r}(s))(r(s) - \bar{r}(s))^T] \\ &= \mathcal{E}\left[\begin{matrix} U(s)W(X - \bar{X}) \\ U(s)W(X - \bar{X})^T \end{matrix} \right] \\ &= \mathcal{E}\left[U(s)W(X - \bar{X})(X - \bar{X})^T W^T U(s)^T \right] \\ &= U(s)W \mathcal{E}\left[(X - \bar{X})(X - \bar{X})^T \right] W^T U(s)^T \\ &= U(s)W \mathcal{V}[X] W^T U(s)^T \end{aligned} \quad \begin{aligned} r(s) - \bar{r}(s) &\stackrel{=}{=} U(s)(WX - Q_0) - U(s)(W\bar{X} - Q_0) \\ &= U(s)W(X - \bar{X}) \\ &= \\ \mathcal{E}[AX+b] &\stackrel{=}{=} A\mathcal{E}[X]+b \\ \mathcal{V}[X] &\stackrel{=}{=} \mathcal{E}[(X - \bar{X})(X - \bar{X})^T] \end{aligned}$$

de lo que se concluye que

$$P_r(s) = U(s)W \mathcal{V}[X] W^T U(s)^T \quad (4-5)$$

Como se mencionó anteriormente, la variabilidad de una curva con respecto a la distribución a la que pertenece se expresa mediante la matriz de covarianza P_0 . Por este motivo, y de acuerdo a (4-6), también se puede expresar $P_r(s)$ como

$$P_r(s) = U(s)WP_0W^T U(s)^T \quad (4-6)$$

4.3.1.1 Desplazamiento cuadrático medio en cada punto

Se define $\rho_0^2(s)$ como la traza de la matriz $P_r(s)$. Este valor representa el desplazamiento cuadrático medio de la curva en cada punto determinado por s

$$\rho_0^2(s) = \text{tr}(P_r(s)) \quad (4-7)$$

La matriz de covarianza de una distribución normal isotrópica es diagonal, y en su diagonal se encuentran los autovalores de la misma. En este caso, todos los autovalores tienen la misma magnitud. Por consiguiente, el valor del desplazamiento cuadrático medio en el caso bidimensional isotrópico es 2λ , donde λ es el valor de los autovalores de $P_r(s)$.

4.3.1.2 Desplazamiento cuadrático medio promedio de una curva

El desplazamiento cuadrático medio a lo largo de toda la curva se llamará $\bar{\rho}_0^2$ y está determinado por la integral a lo largo de la curva de $\rho_0^2(s)$

$$\begin{aligned} \bar{\rho}_0^2 &= \frac{1}{L} \int_0^L \rho_0^2(s) ds = \frac{1}{L} \int_0^L \text{tr}(P_r(s)) ds = \frac{1}{L} \int_0^L \text{tr}(U(s)WP_0W^T U(s)^T) ds \stackrel{\text{tr}(A.B)=\text{tr}(B.A)}{=} \\ &= \frac{1}{L} \int_0^L \text{tr}(P_0W^T U(s)^T U(s)W) ds = \text{tr} \left(P_0W^T \left[\frac{1}{L} \int_0^L U(s)^T U(s) ds \right] W \right) = \\ &= \text{tr}(P_0W^T \mathcal{U}W) = \text{tr}(P_0\mathcal{H}) \end{aligned}$$

De esto se concluye que

$$\bar{\rho}_0^2 = \text{tr}(P_0\mathcal{H}) \quad (4-8)$$

4.3.1.3 Variabilidad de un punto sobre los segmentos normales

Si se quiere ver cuanto desplazamiento ha sufrido un punto en dirección al segmento normal a la curva se debe emplear la expresión

$$D_n(s) = (r(s) - \bar{r}(s)) \cdot \bar{n}(s)$$

Se define $\rho_n^2(s)$ como la variación del desplazamiento de un punto, en la dirección del segmento normal a la curva que pertenece. Este valor se puede calcular como

$$\rho_n^2(s) = \mathcal{V}[D_n(s)] = \mathcal{V}[(r(s) - \bar{r}(s)) \cdot \bar{n}(s)] = \mathcal{V}[r(s) \cdot \bar{n}(s) - \bar{r}(s) \cdot \bar{n}(s)]$$

Dado que la curva $\bar{r}(s)$ se encuentra fija y la curva $r(s)$ es la variable aleatoria,

$$\rho_n^2(s) = \bar{n}(s)^T \mathcal{V}[r(s)] \bar{n}(s)$$

Dado que la variabilidad en un punto es equivalente a $P_r(s)$, se obtiene

$$\rho_n^2(s) = \bar{n}(s)^T \mathcal{V}[r(s)] \bar{n}(s) = \bar{n}(s)^T P_r(s) \bar{n}(s)$$

Se puede suponer que el desplazamiento de un punto a lo largo de una recta normal sigue entonces una distribución $\mathcal{N}(0, \rho_n^2(s))$.

Si se consideran (4-6) y (3-10), también se cumple que

$$\rho_n^2(s) = h(s)^T P_0 h(s) \tag{4-9}$$

Si se considera que la matriz de covarianza es isotrópica, el valor de $\rho_n(s)$ (el desvío estándar) quedará expresado en función de la raíz de cualquier autovalor¹⁵ de $P_r(s)$. Se tendrá entonces

$$\rho_n^2(s) = \begin{bmatrix} n_x(s) & n_y(s) \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} n_x(s) \\ n_y(s) \end{bmatrix} = n_x(s)\lambda n_x(s) + n_y(s)\lambda n_y(s) = \lambda(n_x^2(s) + n_y^2(s))$$

$$\rho_n(s) = \sqrt{\lambda} \sqrt{n_x^2(s) + n_y^2(s)} \quad \equiv \quad \sqrt{\lambda}$$

Las normales son unitarias

El desvío estándar, se puede emplear para estimar la longitud de los segmentos de búsqueda. Una medida razonable, es tomar dos desvíos estándar (para un intervalo de confianza de un 95%). De esta forma los segmentos de búsqueda en cada punto quedarían expresados como $\bar{r}(s) \pm 2\rho_n(s)\bar{n}(s)$.

4.3.1.4 Variabilidad de un punto en la práctica

En esta sección se mostrará con ejemplos cómo la variabilidad del desplazamiento de un punto en la dirección normal ($\rho_n^2(s)$) es empleada para el muestreo usando segmentos normales. Asimismo se intentará dar una idea gráfica de lo que representa este valor.

A modo de ejemplo, considérese una variable aleatoria que sigue una distribución $\mathcal{N}(\bar{X}, \sigma^2)$ en un espacio unidimensional. Si esa distribución se normaliza, el gráfico de la misma es el de la Figura 4-1, donde se pueden observar dos valores característicos: z_0 y $-z_0$.

Si se quiere saber la probabilidad de que una variable aleatoria $X \sim \mathcal{N}(\bar{X}, \sigma^2)$ sea menor que z_0 y mayor que $-z_0$ se debe considerar

¹⁵ Dado que $P_r(s)$ es definida positiva, el valor de sus autovalores es real y positivo.

$$Z = \frac{X - \bar{X}}{\sigma} \sim \mathcal{N}(0,1)$$

$$P(|Z| < z_0) = P\left(\left|\frac{X - \bar{X}}{\sigma}\right| < z_0\right) = P\left(-z_0 < \frac{X - \bar{X}}{\sigma} < z_0\right) = p$$

Quedando finalmente

$$P(-\sigma z_0 + \bar{X} < X < \sigma z_0 + \bar{X}) = p \tag{4-10}$$

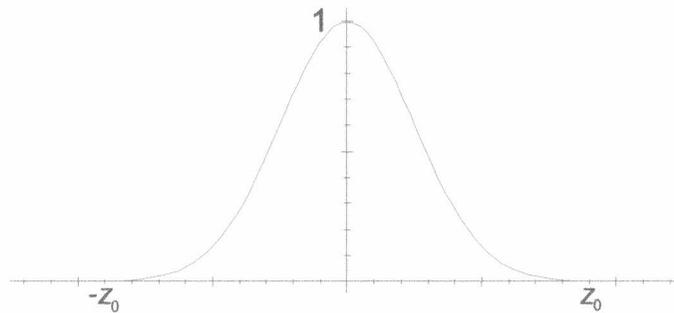


Figura 4-1. Distribución gaussiana normalizada.

En (4-10) se puede ver cómo z_0 y σ , indican el desplazamiento máximo¹⁶ con respecto a la media que puede sufrir una muestra. Si por ejemplo el valor de p es 0.95, entonces sólo el 5% de los puntos muestreados se hallarán desplazados con respecto a la media a una distancia mayor a σz_0 (donde $z_0 = 2$). De esto se concluye que si se tuviera que establecer la longitud de un segmento, tal que partiera desde la media de la distribución y tuviera que alcanzar el 95% de los puntos de la misma, entonces la longitud necesaria que debería tener es 2σ .

Usando lo visto en la sección anterior, el desplazamiento de un punto en una curva en la dirección de la normal a la misma viene dado por $\rho_n(s)$. Dado que el desplazamiento sobre una normal es similar al caso unidimensional se toma $\sigma \equiv \rho_n(s)$, con lo cual la longitud del segmento de búsqueda no necesita superar a $2\rho_n(s)$ para encontrar el punto buscado con un 95% de probabilidad.

El caso que importa a los efectos del presente trabajo, es el bidimensional. Aquí el principio es el mismo, solo que los desplazamientos ocurren en el plano. Por esta razón los valores de X que se evalúan en la función de densidad son bidimensionales, perteneciendo con probabilidad p a un círculo cuyo radio es igual a $|\sigma z_0|$ (Figura 4-2).

¹⁶ Con probabilidad p .

Análogamente al caso unidimensional, si los puntos de la curva $r_f(s)$ siguen una distribución gaussiana cuya media es \bar{r} , entonces cada punto $\bar{r}(s_i)$ tendrá a su correspondiente $r_f(s_i)$ ¹⁷ a una distancia no mayor a σz_0 con probabilidad p (ver Figura 4-3).

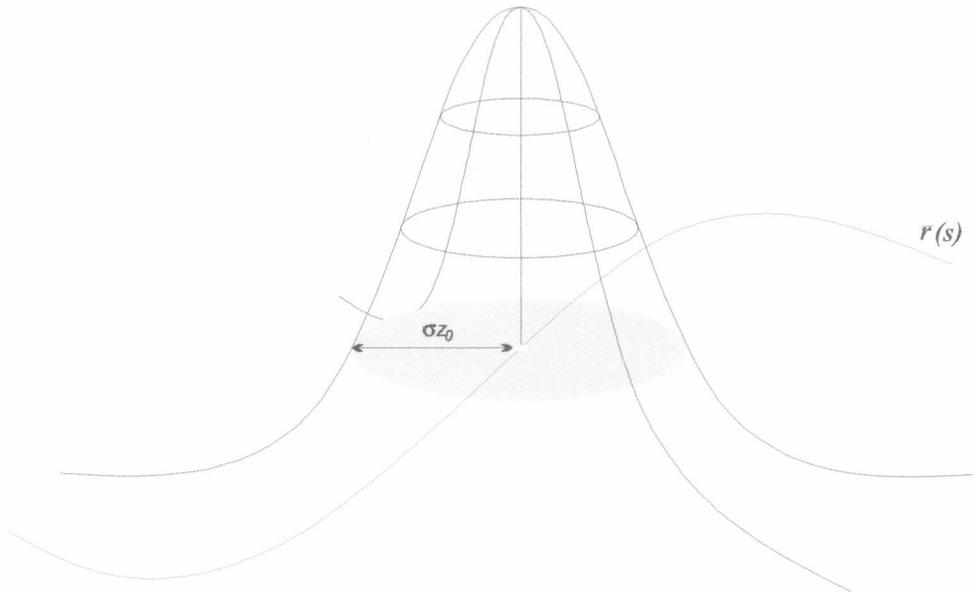


Figura 4-2. En esta figura se observa una distribución gaussiana, cuyo valor medio se ubica sobre la curva $\bar{r}(s)$.

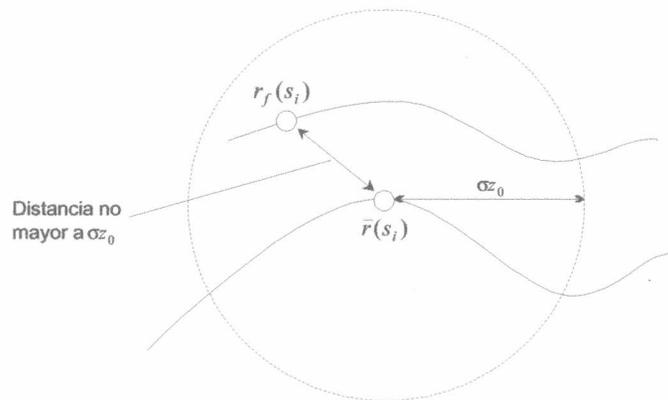


Figura 4-3. En esta figura se puede ver cómo dos puntos que se corresponden se encuentran a una distancia no mayor a σz_0 con una probabilidad que se especifica en función del valor de z_0 . La correspondencia se define mediante el parámetro s ; los dos puntos obtenidos en la evaluación de ambas curvas en el mismo parámetro serán *correspondientes*.

¹⁷ Recuérdese que el contorno reconocido, se construye con la misma cantidad de puntos de control que la curva patrón. De esta manera los puntos de control de ambas curvas, se aparean para su comparación.

Dado que la mayoría de los puntos de $r_f(s)$ se encontrarán dentro del intervalo circular antes mencionado, no tiene sentido buscar a una distancia mayor que el radio de dicho intervalo. Además, pueden ahorrarse muchos cálculos y tiempo de proceso si cada normal restringe su tamaño al diámetro correspondiente al círculo determinado en el punto $\bar{r}(s_i)$. Para más detalles referirse a la sección 4.3.2.

De esta forma, el segmento normal se calcula como

$$\text{segmento normal} = \bar{n}(s) \cdot \kappa \rho_n(s) \quad (4-11)$$

Donde $\kappa = z_0$.

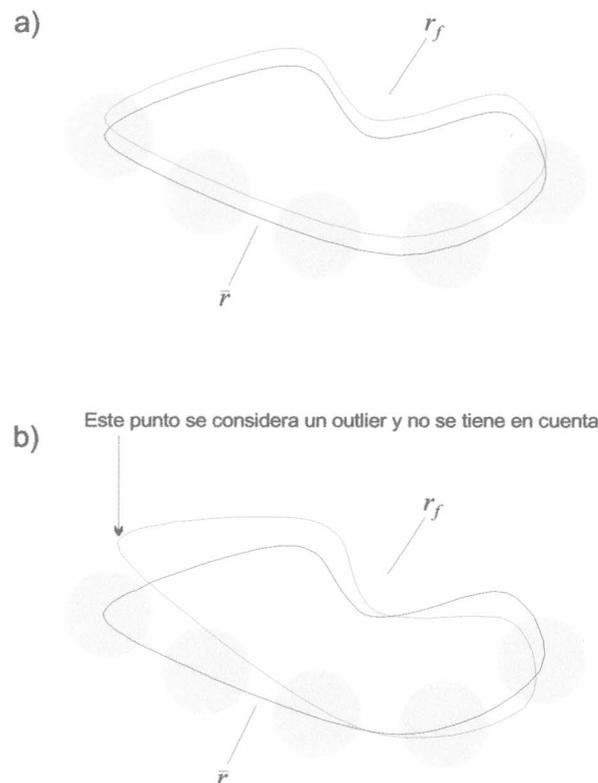


Figura 4-4. En la imagen se puede observar un problema típico que se puede dar por el uso de $\bar{\rho}_0^2$ en lugar de $\rho_n^2(s)$. En **a)** Se tiene una curva $r_f(s)$ en cierta posición. Dado que la ubicación de la curva es muy cercana a la curva patrón la certidumbre es alta; los intervalos circulares (calculados con $\bar{\rho}_0^2$) son pequeños en todos los puntos. En **b)** $r_f(s)$ rota, y debido a que el punto indicado no es encontrado por los segmentos de búsqueda (está más lejos que el radio del intervalo circular), el sistema no reconoce la rotación.

En (4-11) se emplea $\rho_n(s)$ en vez de $\bar{\rho}_0$. Podría usarse sin embargo cualquiera de estos dos valores para calcular la longitud del segmento normal. El uso de $\rho_n^2(s)$ ó $\bar{\rho}_0^2$ dependerá de las circunstancias; emplear $\bar{\rho}_0^2$ permite ahorrar cálculos, ya que su valor se

computa una única vez para todos los puntos de la curva. Sin embargo, los diámetros de los intervalos circulares se equilibrarán mutuamente dando lugar a un círculo *promedio*. Si hay un número importante de puntos donde se registra una alta certidumbre (dando lugar a intervalos circulares pequeños) puede darse que en ciertos casos, se pierdan valores (Figura 4-4). Por otro lado, empleando $\rho_n^2(s)$, la certidumbre se tomará en cuenta de forma independiente en cada $\bar{r}(s_i)$ y consiguientemente, en los puntos donde la incertidumbre es mayor, el intervalo circular también será mayor. Por último y quizás la diferencia más importante, es que, $\bar{\rho}_0^2$ no indica el desplazamiento específicamente en la dirección de los segmentos normales. Esto hace que su precisión se vea degradada cuando se usa en dicho contexto.

La Figura 4-4 ejemplifica un problema que se puede dar por el uso de $\bar{\rho}_0^2$ en lugar de $\rho_n^2(s)$. Aquí el objeto a seguir sufre una rotación y los intervalos de confianza son demasiado pequeños para notarlo. Si el sistema continúa su ejecución, posteriormente se enterará de la rotación, debido a que en iteraciones posteriores se evidenciará la pérdida de certidumbre (generada por la pérdida de puntos). Sin embargo este mecanismo es lento y probablemente para cuando el sistema reaccione, el objeto ya se habrá perdido. Si por el contrario se hubiera empleado $\rho_n^2(s)$, el intervalo de confianza en el punto indicado hubiera sido mayor.

Finalmente se puede decir que tanto $\bar{\rho}_0^2$ como $\rho_n^2(s)$ aportan información sobre la certidumbre y permiten además la eliminación de *outliers* (puntos que se encuentran a una distancia mayor que la determinada por $\kappa\rho_n^2(s)$ o $\kappa\bar{\rho}_0^2(s)$). Esto último se tratará en la siguiente sección, donde se comenta el mecanismo de puerta de validación.

4.3.2 Puerta de validación: Selección de muestras

La *puerta de validación* es un mecanismo para detección y eliminación de *outliers*, puntos que son encontrados por los segmentos normales y que no pertenecen al objeto que se desea seguir. Este método también sirve para poder llevar cuenta de la incertidumbre presente en el sistema.

Anteriormente se explicó que cuando la innovación v_i es grande se puede estar en presencia de un outlier. De acuerdo a la sección 3.2.3,

$$v_i = [r_f(s_i) - \bar{r}(s_i)] \cdot \bar{n}(s_i)$$

Si se suma y resta $r(s_i)$, esta expresión se puede escribir como

$$v_i = [r(s_i) - \bar{r}(s_i)] \cdot \bar{n}(s_i) + [r_f(s_i) - r(s_i)] \cdot \bar{n}(s_i) \quad (4-12)$$

El primer término de (4-12) (el caso general) sigue una distribución $\mathcal{N}(0, \rho_n^2(s))$ (tal como se vio en 4.3.1.3). En el segundo miembro de (4-12), se calcula la diferencia (en la

dirección normal a la curva media) entre un punto de la curva característica ($r_f(s)$) y la curva resultado¹⁸ ($r(s)$).

Considerando la función de densidad de la observación,

$$p(r_f | X) \propto \exp\left[-\frac{N_X}{2\bar{\rho}_f^2} \|r - r_f\|_{\bar{n}}^2\right] \quad (4-13)$$

donde la distancia $\bar{\rho}_f$ puede ser interpretada como el desplazamiento promedio en un punto dado de la curva, debido a errores de observación.

Empleando la discretización presentada en (3-9), se puede reescribir la expresión como

$$p(r_f | X) \propto \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^N [(r_f(s_i) - r(s_i))\bar{n}(s_i)]^2\right] \quad (4-14)$$

donde $\sigma^2 = \frac{\rho_f^2 N}{N_X}$.

Considerando que la distribución de $(r_f(s_i) - r(s_i))\bar{n}(s_i)$ es $\mathcal{N}(0, \sigma^2)$ ¹⁹, se obtiene la distribución seguida por el segundo término de (4-11).

En la Figura 4-5, se puede ver la ubicación del error de medición en el contexto de las curvas.

Se tiene entonces que

$$v_i \sim N(0, \rho^2(s)) \text{ donde } \rho_n^2(s) \leq \rho^2(s) \leq \rho_n^2(s) + \sigma^2$$

Se puede asumir que existe una alta dependencia espacial entre las muestras; esto permitiría utilizar que $\rho(s) = \rho_n(s)$ ²⁰. La distribución de la innovación v_i sugiere que si $|v_i| > \kappa\rho(s)$, entonces la medición $r_f(s)$ se debería descartar por ser un outlier. Tomando el valor de κ igual a 2, se asegura un intervalo de confianza de aproximadamente un 95%.

Lo que se hace en el mecanismo de puerta de validación, es definir la longitud del segmento normal a la curva como $2\rho(s)$. De esta manera, si la innovación supera dicha longitud no se tomará en cuenta la muestra. Este mecanismo requiere una pequeña

¹⁸ Recuérdese que esta curva se obtiene con el vector de control Q , que proviene de la expresión $Q = WX + Q_0$, donde X es la transformación resultado del algoritmo de ajuste.

¹⁹ Donde σ^2 es el error de medición comentado en la sección 3.2.3.

²⁰ Conforme se van haciendo más mediciones, el error en las mismas va disminuyendo. En principio, asumir que las muestras presentan dependencia espacial es razonable, ya que se está siguiendo una estructura que tenderá a mantenerse de una sola pieza.

modificación al algoritmo de ajuste (Figura 3-6). Cabe destacar que si se emplea $\bar{\rho}_0$ en lugar de $\rho_n(s)$, el resultado es similar y sólo se presentan las diferencias antes enunciadas.

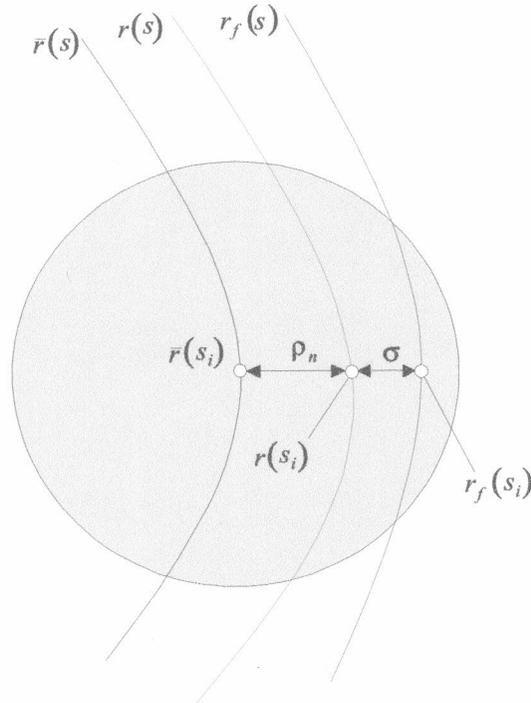


Figura 4-5. En la figura se puede observar cómo los desvíos estándar de las distribuciones sirven para indicar las distancias entre las curvas. También se puede ver en la figura el intervalo circular definido por $2\rho_n$ (el círculo grisado).

De esta forma, cada vez que un punto se encuentre fuera del intervalo determinado por $\kappa\rho_n(s_i)$, no se hará la actualización del vector de observación ni de la matriz de información; esto es equivalente a ignorar la muestra.

El valor de κ (equivalente a z_0) dependerá del grado de confianza que se quiera dar a los puntos encontrados. Si se espera que estos puntos se encuentren muy dispersos, se empleará un valor mayor. Si por el contrario, sólo se quiere considerar valores más cercanos a la media, se empleará un valor de κ más pequeño, haciendo la guarda más restrictiva.

En secciones anteriores, se mencionó que la matriz de información estadística S indica la certidumbre global que se mantiene en el algoritmo. En la Figura 4-6 se puede observar cómo el valor de S depende de la cantidad de muestras que no son consideradas outliers.

<p>Para $i = 0 \dots N$</p> $v_i = (r_f(s_i) - \bar{r}(s_i)) \cdot \bar{n}(s_i)$ $h(s_i)^T = \bar{n}(s_i)^T U(s_i) W$ $\rho_n(s_i) = \sqrt{h(s_i)^T \bar{P} h(s_i)}$ <p>Si $(v_i < \kappa \rho_n(s_i))$ entonces</p> $S_i = S_{i-1} + \frac{1}{\sigma_i^2} h(s_i) h(s_i)^T$ $Z_i = Z_{i-1} + \frac{1}{\sigma^2} h(s_i) v_i$ <p>sino</p> $S_i = S_{i-1}$ $Z_i = Z_{i-1}$
--

Figura 4-6. Modificación del algoritmo de ajuste para permitir la eliminación de outliers.

En el algoritmo clásico de ajuste visto en la sección 3.2, por cada iteración se incrementa la matriz de información S en una cantidad positiva $(\frac{1}{\sigma_i^2} h(s_i) h(s_i)^T)$. El algoritmo propuesto en la Figura 4-6 tiene el mismo funcionamiento, con la diferencia de que la matriz S se incrementa solamente cuando $r_f(s_i)$ se encuentra dentro del intervalo circular definido por $\kappa \rho_n(s_i)$. Cuando la observación $r_f(s_i)$ se encuentra fuera del intervalo de confianza, la matriz S y el vector Z no sufren modificaciones. Si todos los puntos de $r_f(s)$ se encuentran dentro del intervalo circular, la matriz S se hará más grande, indicando una mayor certidumbre y será más fácil predecir cuál será la evolución del movimiento del objeto que se desea seguir. Esto será fuertemente empleado por el filtro de Kalman en el siguiente capítulo.



Capítulo 5: Modelo dinámico probabilístico

En este capítulo se introducen modelos dinámicos probabilísticos para el seguimiento de objetos. Estos modelos probabilísticos son los que se emplearán en el presente trabajo para realizar el seguimiento de un objeto contenido en una secuencia de video. En este capítulo se presentarán los modelos auto-regresivos de primer y segundo orden, y se mostrará cómo cooperan con el filtro de Kalman para lograr realizar el seguimiento.

Para una mejor comprensión del presente capítulo se introducen los siguientes términos

- **Correlación:** La correlación es una medida de la relación entre dos o más variables. Los coeficientes de correlación van desde el -1.0 hasta el 1.0 . Un valor de 0 representa que las variables no se encuentran relacionadas.
- **Cadena de Markov:** Las cadenas de Markov son un tipo de procesos aleatorios. Una cadena de Markov describe un sistema cuyo estado cambia con el tiempo. Los cambios no son completamente predecibles, pero se encuentran gobernados por distribuciones de probabilidad. Estas distribuciones de probabilidad incorporan una especie de estructura de dependencia, donde la distribución condicional de los estados futuros del sistema (dada cierta información sobre los estados pasados) se basa sólo en la información más reciente. Lo que importa a estos procesos, es la predicción del futuro del sistema en su estado actual y no el camino por el cual el sistema llegó al estado presente.
- **Proceso estable:** Un proceso estable, es un proceso que ha alcanzado un estado de equilibrio (*steady state*). Para alcanzar dicho estado, se deben cumplir ciertas restricciones, las cuales se verán en el transcurso del presente capítulo.
- **Estado de *lock* del sistema:** Cuando la cantidad de outliers que encuentra el sistema se halla debajo de cierto umbral, el sistema pasa a un estado de *lock*. En ese estado se asume que el sistema tiene tal certidumbre que resulta innecesaria la observación (el sistema trabaja prediciendo). El sistema puede permanecer unos segundos en el estado de *lock* durante los cuales no es necesaria la observación.

5.1 El modelo probabilístico aplicado al seguimiento

El modelo planteado en (4-2) puede emplearse para realizar un sistema de seguimiento rudimentario. Se puede obtener el valor de X correspondiente al momento actual (denominado \hat{X}_k , donde k representa el número de iteración), basándose en el valor inmediato anterior (\hat{X}_{k-1}). Asumiendo que el algoritmo de seguimiento se encuentra en el k -ésimo paso se tiene (empleando (4-3)) que

$$\hat{X}_k = \max_{X_k} p(X_k | r_f^k) \propto \max_{X_k} (p(r_f^k | X_k) p(X_{k-1} | r_f^{k-1})) \quad (5-1)$$

Donde $p(r_f^k | X_k)$ es la observación en el paso k , y $p(X_{k-1} | r_f^{k-1})$ es el regularizador.

La expresión (5-1) presenta un modelo simple para realizar un seguimiento. Sin embargo, este modelo presenta dos desventajas importantes que lo hacen poco útil para tal fin.

La primer desventaja que presenta este modelo de seguimiento rudimentario, tiene que ver con la poca información brindada por el regularizador, lo que hace que los ajustes a la observación sean mínimos. Básicamente el problema surge cuando sabiendo la posición y forma del objeto en un cuadro, el modelo no puede proveer información acerca de la posición y forma del mismo en el siguiente cuadro. De esta forma, el sistema sólo cuenta con lo que puede observar en cada cuadro, recibiendo poca ayuda del regularizador respecto de cómo será el siguiente cuadro. Dado que no existe forma de extrapolar la forma y posición del objeto entre un cuadro y el siguiente, este modelo sólo funcionará con objetos que se mueven muy lentamente y/o que no se ocluyen²¹. Esta restricción radica en que el sector de interés se genera de acuerdo al valor obtenido en el paso anterior, sin tener en cuenta la historia previa del movimiento. Habiendo fallado en algún paso, se desencadena una secuencia de fallas de la cual el algoritmo no puede recuperarse. Más adelante se mostrará como el filtro de Kalman funciona con un modelo de movimiento que le permite *predecir* la ubicación y forma del objeto a seguir, ubicando el sector de interés correctamente y realizando ajustes más precisos a la observación.

La segunda desventaja que presenta este modelo rudimentario tiene que ver con la acumulación de la matriz de información. Debido a que el resultado del proceso en el paso $k-1$ (\hat{X}_{k-1}) se utiliza como regularizador en el paso k , la matriz de información se ve incrementada a lo largo de las iteraciones. Como se puede observar en la Figura 3-6 (paso 6: $\hat{X} = \bar{X} + (\bar{S} + S)^{-1} Z$), al incrementarse el valor de la matriz de información S , el resultado final adquiere muy poca influencia de la observación (Z). Esto conlleva a que, a partir de cierto momento, en las sucesivas iteraciones el efecto de regularización será cada vez más fuerte, ignorando las observaciones realizadas.

Un esquema de trabajo probabilístico de seguimiento debe proveer una densidad *a priori* que permita salvar los problemas antes enunciados. Esto será tratado en las siguientes secciones.

²¹ Se podría hacer este problema extensible a los sistemas de seguimiento que no realizan estimación o aprendizaje de movimiento.

5.2 Funciones de densidad a priori dinámicas

Dado que la observación se debe limitar a reflejar el contenido de un cuadro de video, no es razonable que se le agregue complejidad adicional. De esto se concluye que la complejidad adicional sobre la que debería apoyarse el sistema de seguimiento, se encontraría en la función de densidad *a priori*. En esta sección se analizará cómo debe ser esta función para salvar problemas tales como la extrapolación de movimiento de un cuadro a otro.

5.2.1 Función de densidad a priori global

Una densidad a priori global²², se puede aprender de un conjunto de muestras. Sin embargo, definir una función de densidad de este estilo no impone una restricción específica en las transformaciones obtenidas en cada paso. Supóngase una partícula que sigue una trayectoria como la que se muestra en la Figura 5-1²³. La trayectoria aprendida permite establecer una función de densidad global, la cual se grafica como a) en la Figura 5-1. Sin embargo, dicha función barre gran parte del recorrido y no aplica una restricción importante a la ubicación de la partícula en el próximo cuadro. Aún cuando la función de densidad a priori global no lo indique, en la Figura 5-1 se puede ver que las posiciones sucesivas de la partícula se hallan muy restringidas. Si por ejemplo se tuviera que decidir cuál es la posición de la partícula en cada instante, lo único que podrá decir la función de densidad a priori global, es que con una alta probabilidad se encuentra dentro de la elipse a). En la figura se puede ver cómo el conjunto de posiciones contenidas en la función de densidad a) es mucho mayor que el conjunto de posiciones reales, representada como el área grisada b), que podría tomar la partícula.

Cuando se considera una secuencia de imágenes, una función de densidad global como $p_0(X)$ no es suficiente para imponer restricciones a \hat{X}_k . Lo que se desea es que el valor de \hat{X}_k cumpla con un conjunto de restricciones que le sean impuestas por las transformaciones de los pasos anteriores, es decir por $\hat{X}_0 \dots \hat{X}_{k-1}$. Esto es razonable, ya que en un cuadro de una secuencia de video, la forma y posición del objeto que se desea seguir están íntimamente relacionados con los que tenía el objeto en el o los cuadros anteriores.

Basándose en una densidad a priori que se construya en cada paso con la información aportada por el resultado de los pasos anteriores, se esperaría obtener una mayor restricción sobre las posibles posiciones que podría tomar la partícula (Figura 5-2).

²² Todos las transformaciones resultado pertenecen a la misma distribución.

²³ Si bien en el ejemplo se muestran solo traslaciones, se puede hacer extensivo a cualquier tipo de deformación que pueda sufrir el objeto a seguir.

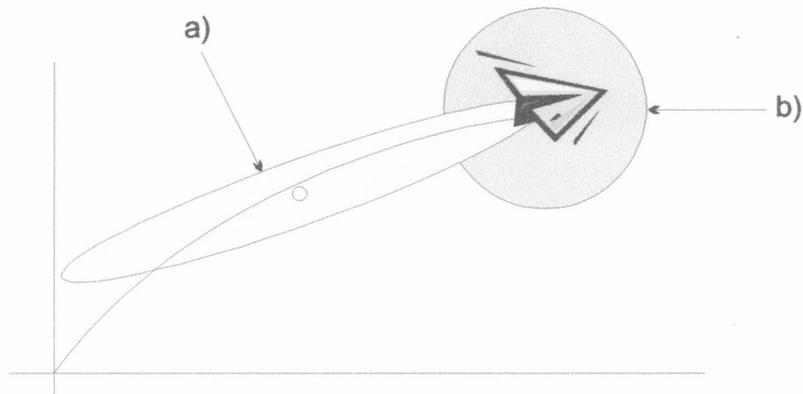


Figura 5-1. Una partícula desplazándose en un plano: a) función de densidad asociada a la media del movimiento de la partícula (la cual ha sido aprendida previamente). b) Conjunto de coordenadas reales que podría tomar la partícula en el siguiente cuadro de la secuencia de video. Se puede ver que a) no restringe de manera importante la ubicación de la partícula en el próximo cuadro.

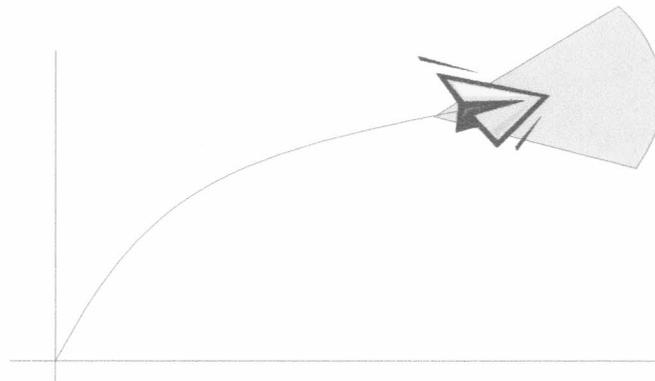


Figura 5-2. Si el movimiento es estudiado previamente, se puede restringir aún más el conjunto de coordenadas que puede tomar la partícula. Compárese con la figura anterior.

La forma de lograr una correlación entre \hat{X}_k y los $\hat{X}_0 \dots \hat{X}_{k-1}$, es confeccionar una función de densidad condicional $p(\hat{X}_k | \hat{X}_0 \dots \hat{X}_{k-1})$. En el modelo presentado en [5], se realiza la siguiente suposición, con el objeto de simplificar el modelo

$$p(X(t_k) | X(t_1) \dots X(t_{k-1})) = p(X(t_k) | X(t_{k-1})) \quad (5-2)$$

El término de la derecha en (5-2), se conoce como *Cadena de Markov* de primer orden.

5.2.2 Proceso de Markov Gausiano de primer orden

Un proceso de Markov de primer orden en el espacio de formas, está definido por la expresión

$$p(X(t_k) | X(t_{k-1})) \propto \exp - \frac{1}{2b^2} \|X(t_k) - X(t_{k-1})\|^2 \quad (5-3)$$

En un espacio de formas de traslaciones en el plano, este proceso representa el movimiento aleatorio de un contorno rígido donde el movimiento de cada uno de sus puntos pueden ocurrir en cualquier dirección isotrópicamente. Este tipo de movimiento aleatorio se conoce como *browniano*.

A los efectos del presente trabajo, este modelo es demasiado aleatorio ya que el objetivo es el seguimiento del cuerpo humano. Con el objeto de restringir los movimientos aleatorios, se hace el agregado de una constante de desplazamiento D . En el espacio de las traslaciones en el plano, D representa el desplazamiento medio en cada paso. De esta forma se modifica (5-3) obteniendo

$$p(X(t_k) | X(t_{k-1})) \propto \exp - \frac{1}{2b^2} \|X(t_k) - X(t_{k-1}) - D\|^2 \quad (5-4)$$

La variable aleatoria que sigue la distribución asociada a la función de densidad (5-4), tiene en el paso n una distribución equivalente a la suma de N gaussianas, lo cual también es una distribución gaussiana. Dada la naturaleza del movimiento desarrollado, en el cual los puntos van tomando valores cada vez más alejados del origen, la varianza tiende a crecer ilimitadamente (ver Apéndice A-Demostración del crecimiento ilimitado de la varianza). Esto último hará que este modelo no sea apropiado cuando se quieran modelar deformaciones junto con traslaciones, ya que el contorno con el que se sigue al objeto deseado se deformará sin ningún tipo de control.

Dado que se quiere que las deformaciones sean forzadas a cumplir con ciertas restricciones, se plantea lo siguiente

$$p(X(t_k) | X(t_{k-1})) \propto \exp - \frac{1}{2b^2} \|(X(t_k) - \bar{X}) - a(X(t_{k-1}) - \bar{X})\|^2 \quad (5-5)$$

Donde con un valor de a entre 0 y 1 se obtiene un movimiento browniano restringido²⁴. Modificando ligeramente (5-5), este modelo permite además hacer explícito la separación existente entre deformaciones y traslaciones²⁵

$$p(X(t_k) | X(t_{k-1})) \propto \exp - \frac{1}{2} \left\| \frac{1}{b^s} E^s \left(\overbrace{(X(t_k) - \bar{X}) - a^s (X(t_{k-1}) - \bar{X})}^{\text{Término que afecta al espacio de traslaciones}} \right) + \frac{1}{b^d} E^d \left(\underbrace{(X(t_k) - \bar{X}) - a^d (X(t_{k-1}) - \bar{X})}_{\text{Término que afecta al espacio de deformaciones}} \right) \right\|^2 \quad (5-6)$$

²⁴ En la siguiente sección se mostrará por que un valor de a entre 0 y 1 provoca que el movimiento alcance un estado de equilibrio.

²⁵ La ortogonalidad de los subespacios, hace que se puedan aplicar procesos de Markov con parámetros distintos en cada uno de ellos.

Donde E^s es la matriz de proyección del espacio de traslaciones y E^d es la matriz de proyección del espacio de deformaciones (ambas definidas en 3.2.1.1).

Tal como se mencionara anteriormente, si a es mayor o igual a 1, entonces se está en presencia de un movimiento *browniano no restringido*. Se puede ver que si en (5-7) $0 < a^d < 1$ y $a^s = 1$ entonces se establece un movimiento *browniano restringido* en el espacio de las deformaciones y un *movimiento browniano no restringido* en el espacio de las traslaciones.

$$p(X(t_k) | X(t_{k-1})) \propto \exp - \frac{1}{2} \left\| \frac{1}{b^s} E^s (X(t_k) - X(t_{k-1})) + \frac{1}{b^d} E^d ((X(t_k) - \bar{X}) - a^d (X(t_{k-1}) - \bar{X})) \right\|^2 \quad (5-7)$$

Esta misma expresión se puede escribir de forma más compacta, en el espacio planar afin, si se considera

$$A = E^s + a^d E^d \quad \text{y} \quad B = (b^s E^s + b^d E^d) \mathcal{H}^{-\frac{1}{2}} \quad (5-8)$$

De esta manera (5-7) se puede escribir como

$$p(X(t_k) | X(t_{k-1})) \propto \exp - \frac{1}{2} \| B^{-1} (X(t_k) - AX(t_{k-1}) - D) \|^2 \quad (5-9)$$

Donde la matriz B es invertible (para una demostración referirse al Apéndice A-Invertibilidad de la matriz B).

Sin embargo, como el modelo de cadena de Markov no es un medio simple e intuitivo para trabajar, se emplearán los procesos auto-regresivos. Estos son una forma más explícita de los modelos de cadena de Markov y serán comentados en la próxima sección.

5.3 Momentos auto-regresivos de primer orden

Trabajar con una expresión como (5-9) no es intuitivo, por lo que en general se prefieren otro tipo de expresiones. A continuación se plantea una expresión más conveniente para trabajar.

Considérese la siguiente expresión

$$X(t_k) = AX(t_{k-1}) + D + Bw_k$$

donde $w_k \sim \mathcal{N}(0, I)$

Si se despeja la variable aleatoria, se obtiene

$$B^{-1} (X(t_k) - AX(t_{k-1}) - D) = w_k$$

Luego la función de densidad de w_k es

$$p(w_k) \propto \exp\left[-\frac{1}{2} \left(B^{-1}(X(t_k) - AX(t_{k-1}) - D) \right)^2\right]$$

Sin embargo y dado que se está trabajando en el espacio de formas, se empleará la norma L_2 en vez de la norma Euclideana. Estas normas tienen un valor similar en la práctica, pero en los casos donde esto no ocurre se recomienda el uso de la norma L_2 . De esta forma

$$\exp\left[-\frac{1}{2} \left(B^{-1}(X(t_k) - AX(t_{k-1}) - D) \right)^2\right] \approx \exp\left[-\frac{1}{2} \| B^{-1}(X(t_k) - AX(t_{k-1}) - D) \|^2\right]$$

Se ha mostrado entonces que la expresión

$$X(t_k) = AX(t_{k-1}) + D + Bw_k \quad (5-10)$$

permite obtener el valor de $X(t_k)$ en función de los demás valores intervinientes en (5-9). Esta expresión es a la vez mucho más intuitiva y más fácil de manipular por un algoritmo.

Reemplazando el desplazamiento medio D por $(I - A)\bar{X}$ se obtiene

$$X(t_k) - \bar{X} = A(X(t_{k-1}) - \bar{X}) + Bw_k \quad (5-11)$$

La expresión (5-11) es la forma estándar de un proceso AR de primer orden, donde w_k es un vector de N_x variables aleatorias, cada una con distribución $\mathcal{N}(0,1)$. Asimismo, w_k y $w_{k'}$ son independientes para $k \neq k'$.

El proceso auto-regresivo puede ser usado para obtener la transformación $X(t_k)$, donde $X(t_k) \sim \mathcal{N}(\hat{X}(t_k), P(t_k))$ para cualquier instante de tiempo t_k . Por otro lado, cuando k tiende a infinito y el proceso es estable²⁶ se cumple que

$$X(t_k) \sim \mathcal{N}(\bar{X}, P_\infty) \quad (5-12)$$

Cálculo de la esperanza

Utilizando que $\hat{X}(t_k) = \mathcal{E}[X(t_k)]$, si se toma la esperanza de (5-11), se obtiene la siguiente ecuación

$$\hat{X}(t_k) - \bar{X} = A(\hat{X}(t_{k-1}) - \bar{X}) \quad (5-13)$$

Se puede ver que \bar{X} satisface (5-13), de donde se concluye que la media de la distribución en el estado de equilibrio es \bar{X} .

²⁶ Una condición necesaria para que el proceso sea estable o que alcance el steady state es que $\|A\|_2 < 1$.

Cálculo de la covarianza

Ya obtenida la media de la distribución en el estado de equilibrio, resta ahora obtener la matriz de covarianza. A continuación se mostrará una expresión que permitirá el cálculo recursivo de la misma:

Considérese la variable aleatoria w_k ; si se despeja dicha variable de la ecuación (5-11) y se calcula la matriz covarianza se obtiene que

$$\mathcal{V}[w_k] = \mathcal{V}[B^{-1}((X(t_k) - \bar{X}) - A(X(t_{k-1}) - \bar{X})))]$$

Dado que $w_k \sim N(0, I)$ y usando que $(I - A)\bar{X} = D$ se tiene

$$\begin{aligned} I &= \mathcal{V}[B^{-1}(X(t_k) - AX(t_{k-1}) - D)] = \\ &= \mathcal{V}[B^{-1}X(t_k) - B^{-1}AX(t_{k-1}) - B^{-1}D] \stackrel{\equiv}{=} \mathcal{V}[AX+C] = \mathcal{V}[X]A^T \\ &= B^{-1}\mathcal{V}[X(t_k)](B^{-1})^T - B^{-1}A\mathcal{V}[X(t_{k-1})]A^T(B^{-1}) \end{aligned}$$

multiplicando en ambos lados por B

$$\begin{aligned} BB^{-1}\mathcal{V}[X(t_k)](B^{-1})^T - BB^{-1}A\mathcal{V}[X(t_{k-1})]A^T(B^{-1})^T &= B \\ \mathcal{V}[X(t_k)](B^{-1})^T - A\mathcal{V}[X(t_{k-1})]A^T(B^{-1})^T &= B \end{aligned}$$

Ahora, empleando la propiedad de que $(B^{-1})^T = (B^T)^{-1}$, se multiplica en ambos miembros por B^T

$$\begin{aligned} \mathcal{V}[X(t_k)](B^T)^{-1}B^T - A\mathcal{V}[X(t_{k-1})]A^T(B^T)^{-1}B^T &= BB^T \\ \mathcal{V}[X(t_k)] - A\mathcal{V}[X(t_{k-1})]A^T &= BB^T \end{aligned}$$

Ahora, llámese $P(t_k)$ a la varianza en el paso k -ésimo

$$\begin{aligned} P(t_k) - AP(t_{k-1})A^T &= BB^T \\ P(t_k) &= AP(t_{k-1})A^T + BB^T \end{aligned}$$

Como resultado, se ha hallado una fórmula recursiva para el cálculo de la varianza, la cual depende del paso k . Esta fórmula es

$$P(t_k) = AP(t_{k-1})A^T + BB^T \quad (5-14)$$

Si se considera el estado de equilibrio, se puede ver que $P_\infty \equiv \lim_{k \rightarrow \infty} P(t_k)$ es un punto fijo de (5-14), con lo que

$$P_{\infty} = AP_{\infty}A^T + BB^T \quad (5-15)$$

Más adelante se mostrará qué condiciones deben darse para que (5-15) alcance un punto fijo.

Restricciones de los procesos auto-regresivos de primer orden

Los procesos auto-regresivos son estocásticos y pueden modelar una amplia gama de movimientos realizados por un objeto, simplemente variando los parámetros A y B . Estos procesos pueden trabajar con cambios en la posición y forma, y pueden imponer fuertes restricciones cuando las propias del esquema de seguimiento son débiles. Sin embargo suelen ser muy restrictivos en ciertos aspectos, sobre todo en lo que concierne a las traslaciones. Los procesos AR de primer orden proveen mecanismos para trabajar con desplazamientos, pero éstos están prefijados en una determinada dirección (D en (5-10)). Dado que no es posible seguir un objeto que no tenga una dirección definida, el modelo no aporta ayuda alguna cuando el seguimiento se realiza sobre cuerpos que realizan movimientos indefinidos o cíclicos. De hecho, el modelo de AR de primer orden es incapaz de modelar cualquier tipo de comportamiento cíclico u oscilatorio (no restringiéndose esta falla únicamente a las traslaciones). Si se quiere seguir por ejemplo las deformaciones presentes en un corazón, el modelo AR de primer orden no podrá realizar aporte alguno para ayudar al sistema de seguimiento.

5.4 Momentos auto-regresivos de segundo orden

Los momentos AR de segundo orden complementan los requerimientos adicionales (movimiento no determinístico y oscilatorio) de los que carecía el modelo AR mostrado en la sección anterior. A diferencia de los momentos AR de primer orden, los momentos AR de segundo orden reciben la influencia de dos predecesores. Su expresión en el espacio de formas es

$$X(t_k) - \bar{X} = A_2(X(t_{k-2}) - \bar{X}) + A_1(X(t_{k-1}) - \bar{X}) + B_0 w_k \quad (5-16)$$

Que por simplicidad se puede expresar también como

$$\mathcal{X}(t_k) - \bar{\mathcal{X}} = A(\mathcal{X}(t_{k-1}) - \bar{\mathcal{X}}) + B w_k \quad (5-17)$$

donde

$$\mathcal{X}(t_k) = \begin{pmatrix} X(t_{k-1}) \\ X(t_k) \end{pmatrix}, \quad \bar{\mathcal{X}} = \begin{pmatrix} \bar{X} \\ \bar{X} \end{pmatrix}, \quad A = \begin{pmatrix} 0 & I \\ A_2 & A_1 \end{pmatrix} \quad y \quad B = \begin{pmatrix} 0 \\ B_0 \end{pmatrix} \quad (5-18)$$

En (5-18) se presentan A_2 , A_1 y B_0 . Estos valores serán presentados más adelante, ya que por el momento basta con decir que permiten establecer los parámetros (análogos a los de los AR de primer orden) que describen las transformaciones sufridas por el objeto a seguir.

En (5-18) se está empleando una cadena de Markov de segundo orden, ya que el vector de formas en el paso k , depende de los valores obtenidos en los pasos $k-1$ y $k-2$.

La esperanza y la covarianza de \mathcal{X} en el paso k son

$$\hat{\mathcal{X}}(t_k) = \mathcal{E}[\mathcal{X}(t_k)] \quad \text{y} \quad \mathcal{P}(t_k) = \mathcal{V}[\mathcal{X}(t_k)] \quad (5-19)$$

donde

$$\mathcal{P}(t_k) = \begin{bmatrix} P''(t_k) & P'(t_k)^T \\ P'(t_k) & P(t_k) \end{bmatrix} \quad (5-20)$$

En (5-20) el término más interesante es $P(t_k)$, que determina la matriz de covarianza de $X(t_k)$ en el paso k -ésimo. $P''(t_k)$ representa la matriz de covarianza de $X(t_{k-1})$ y $P'(t_k) = \mathcal{E}[(X(t_k) - \hat{X}(t_k))(X(t_{k-1}) - \hat{X}(t_{k-1}))^T]$. Tanto $P'(t_k)$ como $P''(t_k)$ son mantenidas para llevar la historia requerida por los procesos AR de segundo orden.

Las ecuaciones recursivas para obtener la media y la matriz de covarianza son análogas a las del modelo AR de primer orden

$$\hat{\mathcal{X}}(t_k) - \bar{\mathcal{X}} = A(\hat{\mathcal{X}}(t_{k-1}) - \bar{\mathcal{X}}) \quad (5-21)$$

y

$$\mathcal{P}(t_k) = A\mathcal{P}(t_{k-1})A^T + BB^T \quad (5-22)$$

5.5 Parámetros dinámicos del modelo

Según se ha visto en secciones anteriores, los modelos auto-regresivos incluyen entre sus parámetros a las matrices A y B . Para el caso unidimensional, estas matrices tienen la siguiente forma

$$A = \begin{bmatrix} 0 & 1 \\ a_2 & a_1 \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 0 \\ b_0 \end{bmatrix} \quad (5-23)$$

donde

$$a_2 = -e^{-2\beta\tau}, \quad a_1 = 2e^{-\beta\tau} \cos(2\pi f\tau) \quad \text{y} \quad b_0 = \bar{\rho} \sqrt{1 - a_2^2 - a_1^2 - 2 \left(\frac{a_2 a_1^2}{1 - a_2} \right)} \quad (5-24)$$

En el caso particular de considerar un espacio de formas, vale que $A_1 = a_1 I$, $A_2 = a_2 I$ y

$$B_0 = \frac{b_0}{\sqrt{N_x}} \mathcal{H}^{\frac{1}{2}}.$$

En (5-24) f representa la frecuencia inicial del movimiento y β representa una tasa de amortiguación (*damping rate*), que se aplica a dicha frecuencia alterándola y dando por resultado el gráfico que se ve en la Figura 5-3.

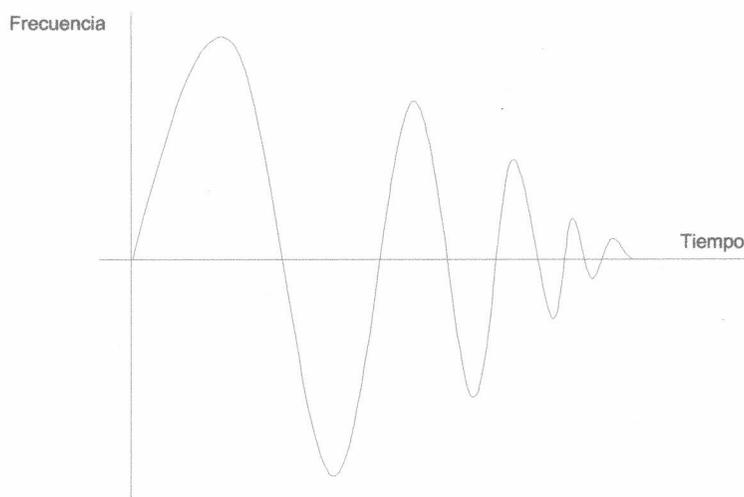


Figura 5-3. Curva en la que se ve cómo la frecuencia disminuye con el tiempo, cuando se le aplica un tasa de amortiguación.

El conjunto de valores que toman f y β establece la secuencia de transformaciones que serán devueltas por el proceso AR de segundo orden. Dentro de ese conjunto, existen algunos casos particulares:

En el caso particular donde $\beta = 0$, el modelo permite la caracterización de un movimiento cíclico de amplitud constante, ya que la frecuencia no disminuye nunca. Además, el movimiento no alcanza nunca un estado de equilibrio (*steady state*). Para ver una demostración, referirse al Apéndice A-Condiciones sobre β para alcanzar el estado de equilibrio.

Otro caso particular ocurre cuando $\beta = f = 0$, donde el movimiento no será cíclico. Una partícula que siga un movimiento así definido, cumplirá con la propiedad de moverse (transformarse) a velocidad constante. Este hecho se puede ver si se reemplazan los valores de β y f en (5-24) y se emplea el resultado en (5-16)²⁷ obteniendo

$$X(t_k) - \bar{X} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & -1 \end{bmatrix} (X(t_{k-2}) - \bar{X}) + \begin{bmatrix} 2 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & 2 \end{bmatrix} (X(t_{k-1}) - \bar{X})$$

Si se considera que $X(t_0) = 0$ y $X(t_1) = \hat{X}(t_1)$ (el valor de la transformación correspondiente al primer cuadro de video²⁸), se puede mostrar que la secuencia queda:

²⁷ Considerando un espacio de formas.

²⁸ Esta decisión es arbitraria pudiéndose emplear aquí cualquier transformación que se desee. En este caso lo que se ha hecho es mantener consistencia con la implementación, donde el valor en t_1 es la transformación obtenida luego de procesar el primer cuadro de video.

$$\begin{aligned}X(t_0) &= 0 \\X(t_1) &= 2\hat{X}(t_1) \\X(t_2) &= 0 + 2.2\hat{X}(t_1) = 4\hat{X}(t_1) \\X(t_3) &= -2\hat{X}(t_1) + 2.4\hat{X}(t_1) = 6\hat{X}(t_1) \\&\dots \\X(t_k) &= 2.k.\hat{X}(t_1)\end{aligned}$$

Se puede ver que las componentes de $X(t_k)$ crecen linealmente. En particular si las componentes de las traslaciones crecen linealmente, se está en presencia de un objeto que se encuentra desplazándose con velocidad uniforme.

5.6 El estado de equilibrio

Los procesos auto-regresivos pueden encontrar, si se cumplen ciertas condiciones, un punto de equilibrio o *steady state*. Una vez llegado este punto, las transformaciones X que se van obteniendo sucesivamente pertenecerán a la misma distribución (5-12).

Como se mencionó anteriormente, una de las condiciones para que un proceso auto-regresivo de primer o segundo orden llegue a un estado de equilibrio, es que exista un punto fijo en la ecuación (5-14) o (5-22)²⁹ respectivamente.

En el estado de equilibrio, la matriz de covarianza P tiene un valor constante; una vez calculado, todas las operaciones en las que esta matriz intervenga, como por ejemplo en el cálculo de la longitud de los segmentos normales en la ecuación (4-9), no requerirán que se los calcule nuevamente. Hasta ahora, recalcular la matriz no era necesario ya que sólo se consideraba una única distribución³⁰ correspondiente al único cuadro de video que se estaba analizando. Dado que en cada cuadro se genera una nueva distribución, en el modelo dinámico habrá tantas matrices de covarianza como intervalos de tiempo considerados³¹. En el modelo dinámico, si el proceso que se está considerando ha alcanzado un estado de equilibrio, entonces no se requiere calcular P en cada cuadro debido a que (5-14) ó (5-22) ha alcanzado un punto fijo. No recalcular P en cada cuadro ahorra mucho tiempo de procesamiento y permite que el sistema funcione más rápido. Asimismo, es útil poseer precalculada la matriz P correspondiente al estado de equilibrio. De esta forma el sistema se inicializa con la matriz covarianza del estado de equilibrio y no se pierde tiempo de ejecución buscándola.

²⁹ Esto es equivalente a decir que la matriz de covarianza converge. Para alcanzar este estado, se deben cumplir algunas condiciones iniciales sobre los parámetros del modelo auto-regresivo. Estas condiciones fueron mostradas en la sección anterior.

³⁰ Dicha distribución es $\mathcal{N}(\bar{X}, S^{-1})$. Para más detalles referirse a la sección 4.2-Distribución de transformaciones: el algoritmo de ajuste.

³¹ En el caso del modelo dinámico se considera que cada intervalo de tiempo tiene asociado un cuadro de video.

Un problema que se presenta cuando alcanza el estado de equilibrio, es que el modelo pierde sensibilidad. Esto ocurre porque en el estado de equilibrio no se supone que el sistema realice observaciones del objeto que se encuentra siguiendo, sino que se debe basar en el modelo de movimiento escogido para realizar el seguimiento. Esto determina que si ocurre algún cambio súbito en el movimiento del objeto que se está siguiendo y se pierde el *lock* sobre el mismo, el sistema corre riesgo de desestabilizarse. Es por esto que el estado de equilibrio debe ser usado con discreción, evitando extender su uso por un tiempo prolongado. De acuerdo a [6], para que un modelo de seguimiento en estado de equilibrio no se desestabilice, se deben cumplir las siguientes restricciones:

1. La secuencia de posiciones en las que se muestrea la curva, es la misma de iteración a iteración. Es decir, la secuencia de puntos de la curva que son muestreados y a partir de los cuales se trazan los segmentos normales es la misma en el conjunto de iteraciones que pertenecen al mismo estado de equilibrio.
2. El contorno del objeto se halla en estado de *lock*.
3. Todos los parámetros del modelo han sido configurados.
4. Las rectas normales no han rotado. En la práctica esto exige que la traslación sea pequeña.

En resumen, el estado de equilibrio permite que un modelo de seguimiento mantenga en estado de *lock* a un objeto por unos pocos segundos.

De acuerdo a lo que se concluye al final de la sección 5.2.2, el modelo de cadena de Markov que da origen a los momentos AR empleados en este trabajo, obtiene una convergencia en el espacio de las deformaciones pero no en el espacio de las traslaciones. Eso se estableció considerando que $a^s = 1$ y $0 < a^d < 1$ ³². De esta forma, considerando por separado el cálculo de matrices de covarianza para las deformaciones y para las traslaciones, la ecuación $\mathcal{P}(t_k) = A^d \mathcal{P}(t_{k-1}) (A^d)^T + BB^T$ será convergente y la ecuación $\mathcal{P}(t_k) = A^s \mathcal{P}(t_{k-1}) (A^s)^T + BB^T$ será divergente. Esto hará que $\mathcal{P}(t_k) = A \mathcal{P}(t_{k-1}) A + BB^T$ sea divergente. Por esta razón no se alcanza un estado de equilibrio en los procesos intervinientes en la presente implementación.

5.7 Aprendizaje automático de los parámetros

Los parámetros de los procesos auto-regresivos pueden establecerse a mano para representar movimientos sencillos, por ejemplo movimientos con velocidad constante y oscilaciones³³. Se pueden realizar pruebas manuales para refinar estos parámetros y

³² $\rho(A^s) > 1$ y $0 < \rho(A^d) < 1$ si se considera un espacio multidimensional.

³³ El filtro de Kalman (que se verá más adelante) siempre requiere de un modelo de estimación de movimiento. En el sistema implementado, el modelo de estimación de movimiento cuando no se ha realizado el aprendizaje es el de velocidad constante. Este no representará las transformaciones que sufrirá el objeto en el caso general. Por este motivo, hasta que no se haya ejecutado el

conseguir modelar movimientos más complejos, pero es un trabajo difícil y poco sistemático, especialmente si se utilizan espacios de formas con gran cantidad de dimensiones que pueden llegar a tener alto acoplamiento entre las mismas.

Lo ideal cuando la secuencia de video considerada presenta transformaciones complejas³⁴ del objeto, es aprender los parámetros utilizando secuencias de aprendizaje. Primero se toma un sistema de seguimiento cuyos parámetros han sido establecidos a mano y se lo emplea en una secuencia de entrenamiento que contenga un objeto fácil de seguir (el objeto realiza movimientos lentos y no se ocluye con otros objetos)³⁵. Una vez que los parámetros han sido aprendidos, se pueden utilizar los mismos para construir un sistema de seguimiento más robusto y que esté especialmente ajustado al tipo de movimiento que se desea seguir. Este nuevo sistema de seguimiento se puede utilizar para seguir un objeto con algún propósito específico o se puede utilizar en una secuencia de entrenamiento más compleja³⁶. En la Figura 5-4 se puede observar el ciclo de aprendizaje y seguimiento.

En términos matemáticos, el problema es estimar los coeficientes de los momentos AR de segundo orden A_1, A_2, \bar{X}, B_0 (ver ecuación (5-16)) que mejor modelen el movimiento de la secuencia de entrenamiento X_1, \dots, X_M donde $X_k \equiv X(t_k)$.

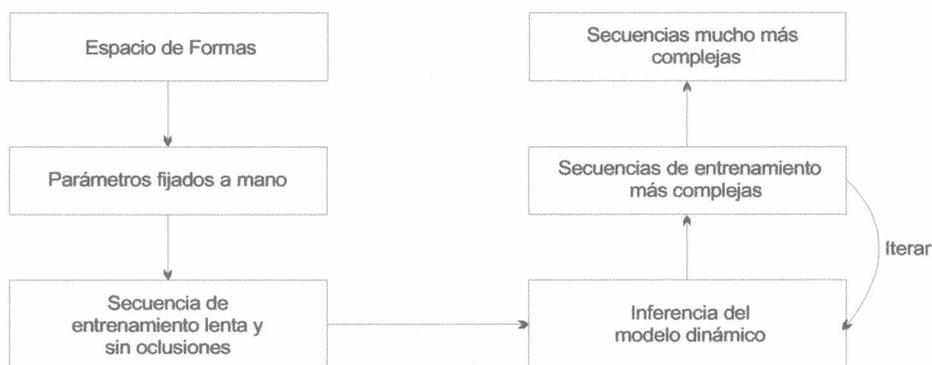


Figura 5-4 Aprendizaje iterativo de los parámetros dinámicos

algoritmo de aprendizaje de parámetros dinámicos y obtenido dichos valores, se asumirá que el sistema no cuenta con un modelo de estimación de movimiento del objeto que pretende seguir.

³⁴ Una transformación compleja es aquella que no podría ser realizada por el sistema, de no contar éste con algún modelo de estimación de movimiento. Las causas más comunes por las cuales estas transformaciones aparecen son movimientos rápidos, oclusiones y escenas con mucho ruido de fondo.

³⁵ Recuérdese que la idea de tener un modelo de movimiento es ayudar al sistema de seguimiento para que no pierda al objeto. No teniendo aún un modelo de movimiento, si la secuencia de video empleada para aprender es muy rápida o compleja, el sistema no podrá seguir al objeto y los parámetros dinámicos no se aprenderán correctamente.

³⁶ De esta forma se puede ir ajustando cada vez más los parámetros dinámicos del modelo.

5.7.1 Aprendizaje de los parámetros dinámicos en una dimensión

Se presentará primero el algoritmo de aprendizaje para el caso de una partícula desplazándose en un espacio unidimensional. Bajo estas consideraciones, los coeficientes dinámicos A_1, A_2 y B_0 se transforman en escalares a_1, a_2 y b_0 . Por simplicidad, se asumirá que $\bar{x} = 0$, de forma tal que, por (5-16), la cantidad $x_k - a_2 x_{k-2} - a_1 x_{k-1}$ es una variable $b_0 w_k$ independiente, normal y con media igual a cero cuya varianza b_0^2 es desconocida. En la Figura 5-5 se esboza el algoritmo utilizado para estimar los valores necesarios (para una demostración del origen del algoritmo, referirse al Apéndice A-Demostración del origen del algoritmo de aprendizaje automático de parámetros).

El problema se puede expresar en términos de una función de *probabilidad logarítmica* definida como

$$L(x_1, \dots, x_k | a_1, a_2, b_0) \equiv \log p(x_1, \dots, x_k | a_1, a_2, b_0) + cte \quad (5-25)$$

donde, debido a que los w_k son independientes,

$$p(x_1, \dots, x_k | a_1, a_2, b_0) \propto \prod_k p_{b_0 w_k}(x_k - a_2 x_{k-2} - a_1 x_{k-1}) \quad (5-26)$$

Usando que $b_0 w_k$ sigue una distribución normal para todo k , se obtiene que

$$L(x_1, \dots, x_k | a_1, a_2, b_0) = -\frac{1}{2b_0^2} \sum_{k=3}^M (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2 - (M-2) \log b_0 + cte \quad (5-27)$$

Problema de aprendizaje dinámico en el caso unidimensional

Dado un conjunto de entrenamiento $\{x_1, \dots, x_M\}$ proveniente de una secuencia de imágenes, aprender los parámetros a_1, a_2 y b_0 de un proceso auto regresivo de segundo orden que describa la dinámica de la partícula moviéndose.

Algoritmo

1. Primero, se calculan los coeficientes de auto-correlación r_{ij} para $i, j = 0, 1, 2$

$$r_{ij} = \sum_{k=3}^M x_{k-i} x_{k-j} \quad \text{para } i, j = 0, 1, 2$$

2. Resolviendo simultáneamente las siguientes ecuaciones se pueden estimar los valores de \hat{a}_1 y \hat{a}_2

$$\begin{aligned} r_{02} - \hat{a}_2 r_{22} - \hat{a}_1 r_{12} &= 0 \\ r_{01} - \hat{a}_2 r_{21} - \hat{a}_1 r_{11} &= 0 \end{aligned}$$

3. Se estima el coeficiente de covarianza como

$$\hat{b}_0^2 = \frac{1}{M-2} (r_{00} - \hat{a}_2 r_{02} - \hat{a}_1 r_{10})$$

Figura 5-5. Algoritmo de aprendizaje de los parámetros dinámicos en una dimensión.

Los coeficientes a_1, a_2 y b_0 se obtienen maximizando la función L , y esto se obtiene fácilmente ya que L es cuadrática. Los valores estimados \hat{a}_1 y \hat{a}_2 están determinados por la minimización de

$$\sum_{k=3}^M (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2 \tag{5-28}$$

cuyas derivadas con respecto a a_1 y a_2 se igualan a cero para obtener \hat{a}_1 y \hat{a}_2 . Esto puede observarse en el paso 2 del algoritmo presentado en la Figura 5-5.

Ahora, a_1 y a_2 pueden tratarse como constantes en L y fijarse en sus valores estimados. L puede maximizarse con respecto a b_0 para obtener \hat{b}_0

$$\hat{b}_0^2 = \frac{1}{M-2} \sum_{k=3}^M (x_k - \hat{a}_2 x_{k-2} - \hat{a}_1 x_{k-1})^2 \tag{5-29}$$

Este valor se calcula en el paso 3 del algoritmo de la Figura 5-5 utilizando los coeficientes de auto correlación.

5.7.2 Aprendizaje de los parámetros dinámicos en el espacio de formas

El algoritmo de aprendizaje para el caso multidimensional sigue los mismos principios que el caso unidimensional. Sin embargo, a diferencia del caso unidimensional, la media \bar{X} no se asumirá conocida y deberá ser aprendida.

La función de probabilidad logarítmica para la distribución normal multidimensional se escribe como

$$L(X_1, \dots, X_M | A_1, A_2, C, \bar{X}) = -\frac{1}{2} \sum_{k=3}^M \left| B_0^{-1} (X'_k - A_2 X'_{k-2} - A_1 X'_{k-1}) \right|^2 - (M-2) \log \det B_0 \quad (5-30)$$

donde

$$X'_k = X_k - \bar{X} \quad \text{y} \quad C = B_0 B_0^T$$

El problema es estimar A_1 , A_2 , \bar{X} y C maximizando la probabilidad logarítmica L . Este no es un problema lineal ya que a diferencia del caso unidimensional, la media \bar{X} tiene que ser estimada. La no linealidad (los términos $A_2 \bar{X}$ y $A_1 \bar{X}$ que aparecen dentro de $|\cdot|^2$) puede ser eliminada usando la forma alternativa de procesos auto regresivos (5-10) adaptada para AR de segundo orden, en la cual

$$D = (I - A_2 - A_1) \bar{X}$$

De esta forma, la función de probabilidad se reescribe como

$$L(X_1, \dots, X_M | A_1, A_2, C, D) = -\frac{1}{2} \sum_{k=3}^M \left| B_0^{-1} (X_k - A_2 X_{k-2} - A_1 X_{k-1} - D) \right|^2 - (M-2) \log \det B_0 \quad (5-31)$$

la cual es cuadrática en A_1 , A_2 y D .

Maximizando la probabilidad logarítmica L se obtiene el algoritmo de la Figura 5-6, que estima los parámetros dinámicos A_1 , A_2 , D y C . Se puede observar que este algoritmo es una generalización del algoritmo de una dimensión presentado anteriormente.

Problema de aprendizaje dinámico en el espacio de formas

Dado un conjunto de entrenamiento de contornos $\{X_1, \dots, X_M\}$ provenientes de una secuencia de imágenes, aprender los parámetros A_1, A_2, B_0 y \bar{X} de un proceso auto regresivo de segundo orden que describa la dinámica del contorno en la secuencia.

Algoritmo

1. Primero, se calculan las sumas $R_i, i = 0, 1, 2$ y los coeficientes de auto-correlación R_{ij} y $R'_{ij}, i, j = 0, 1, 2$

$$R_i = \sum_{k=3}^M X_{k-i}, \quad R_{ij} = \sum_{k=3}^M X_{k-i} X_{k-j}^T, \quad R'_{ij} = R_{ij} - \frac{1}{M-2} R_i R_j^T$$

2. Los parámetros estimados \hat{A}_1, \hat{A}_2 y \hat{D} están determinados por

$$\begin{aligned} \hat{A}_2 &= (R'_{02} - R'_{01} R'^{-1}_{11} R'_{12}) (R'_{22} - R'_{21} R'^{-1}_{11} R'_{12})^{-1} \\ \hat{A}_1 &= (R'_{01} - \hat{A}_2 R'_{21}) R'^{-1}_{11} \\ \hat{D} &= \frac{1}{M-2} (R_0 - \hat{A}_2 R_2 - \hat{A}_1 R_1) \end{aligned}$$

3. La media \bar{X} se estima como

$$\hat{\bar{X}} = (I - \hat{A}_2 - \hat{A}_1)^{-1} \hat{D}$$

4. El coeficiente de covarianza B_0 se estima como la raíz cuadrada $\hat{B}_0 = \sqrt{\hat{C}}$ donde

$$\hat{C} = \frac{1}{M-2} (R_{00} - \hat{A}_2 R_{20} - \hat{A}_1 R_{10} - \hat{D} R_0^T)$$

Figura 5-6. Algoritmo de aprendizaje de los parámetros dinámicos en el espacio de formas.

5.8 El filtro de Kalman

El filtro de Kalman es un método estadístico que permite agregar un modelo de estimación de movimiento a las observaciones realizadas en cada cuadro de video. De esta forma se logra que las observaciones sean mucho más precisas y a la vez se ajusten de acuerdo a la estimación de movimiento, logrando finalmente un sistema mucho más estable. Para el modelo de estimación de movimiento, el filtro de Kalman emplea momentos auto-regresivos de segundo orden, generando una distribución gaussiana en cada iteración. A partir de esa distribución, el método devuelve en cada iteración, una variable aleatoria perteneciente a la misma. Esta variable aleatoria se arma sobre la base de una predicción (una estimación de la variable aleatoria X , generada a partir de la estimación de movimiento) y una observación realizada en el cuadro de video correspondiente a la iteración. Este método, se compone de tres grandes pasos:

- **Predicción:** En este primer paso se intenta predecir cual será la forma y posición del contorno del objeto a seguir, en la presente iteración. Esta es la parte estocástica del método.
- **Medición:** En este paso, se realiza la observación de un cuadro de video. Esta parte del método, aporta la componente determinística del resultado.
- **Asimilación:** En este paso se realizan ajustes a la predicción realizada en el primer paso, de acuerdo a las valores observados en el segundo paso. Hecho esto, se devuelve el resultado de la iteración, en forma de un vector de transformación del espacio de formas.

5.8.1 Predicción

El pseudo código correspondiente al paso de predicción del filtro de Kalman, se puede observar en la Figura 5-7

$$\begin{aligned} 1. \tilde{P}^n(t_k) &= P(t_{k-1}) \\ 2. \tilde{P}'(t_k) &= A_2 P'^T(t_{k-1}) + A_1 P(t_{k-1}) \\ 3. \tilde{P}(t_k) &= A_2 P^n(t_{k-1}) A_2^T + A_1 P'(t_{k-1}) A_2^T + A_2 P'^T(t_{k-1}) A_1^T + A_1 P(t_{k-1}) A_1^T + B_0 B_0^T \\ 4. \tilde{X}'(t_k) &= \hat{X}(t_{k-1}) \\ 5. \tilde{X}(t_k) &= A_2 \hat{X}'(t_{k-1}) + A_1 \hat{X}(t_{k-1}) + (I - A_2 - A_1) \bar{X} \end{aligned}$$

Figura 5-7. Paso encargado de la predicción en el filtro de Kalman.

A continuación se describirá lo que ocurre en cada línea del pseudo código de la Figura 5-7:

1-2-4. En estas líneas se guardan los parámetros de la iteración anterior, de forma de poder emplearlos en la iteración actual. Cabe recordar que al trabajar con procesos auto-regresivos de segundo orden, se está empleando la historia de las dos iteraciones anteriores.

3. En esta línea, se realiza el cálculo de la matriz de covarianza (de acuerdo a (5-22)). Tal como se mencionó anteriormente, esta matriz será empleada por el algoritmo de ajuste para el cálculo de la variabilidad posicional en cada punto ($P_r(s)$), así como la varianza de la distribución de los puntos pertenecientes al segmento normal ($\rho_n^2(s)$). Esta matriz de covarianza que se calcula, es equivalente a la matriz P_0 mencionada anteriormente.

5. En esta línea, se calcula la estimación de la transformación X resultado final del algoritmo. Esta transformación estimada, permitirá la confección del sector de interés que será usado por el algoritmo de ajuste en la parte de medición. Al igual que en la estimación de la matriz de covarianza, se emplean los AR de segundo orden.

En esta parte del algoritmo, es fundamental el hecho de que los parámetros del modelo (A_1, A_2, B_0) correspondan con la dinámica de aquello que se quiere seguir. Estos

parámetros pueden ser aprendidos automáticamente (sección 5.7.2) o pueden ser fijados a mano. De esta forma, la parte de estimación del filtro de Kalman actúa como la función de densidad *a priori* antes buscada, la cual permite hacer una extrapolación del movimiento entre un cuadro y otro.

Si el aprendizaje (automático o manual) se realiza sobre la base de un objeto cuyo comportamiento es diferente al que se está siguiendo, todo el paso predictivo será inútil. Esto ocurre debido a que se predice sobre la base de una dinámica que no es la que sigue el objeto actual.

Finalmente, si el sistema alcanza el estado de equilibrio (*steady state*), la matriz $\tilde{P}(t_k)$ será la misma de iteración en iteración. En un sistema con esta propiedad, dicha matriz podrá ser calculada de antemano, ahorrando su cálculo en tiempo de ejecución (correspondiente a las líneas 1, 2 y 3). De todas formas, como se mencionó anteriormente, la dinámica de los objetos que serán seguidos por la presente implementación no contempla un estado de equilibrio.

5.8.2 Medición

El paso correspondiente a la medición del filtro de Kalman tiene como protagonista al algoritmo de ajuste enunciado en la sección 3.2.2. Este algoritmo recibe la matriz de covarianza $\tilde{P}(t_k)$ y la transformación predicha $\tilde{X}(t_k)$ como parámetros de entrada.

Dentro del algoritmo, para definir el sector de interés a partir del cual se trazarán las rectas normales, se empleará la curva definida por $W\tilde{X}(t_k) + Q_0$ ³⁷. Anteriormente se mencionó que para la definición del sector de interés, se empleaba la curva patrón del sistema. Sin embargo, si el objeto va cambiando su posición y forma en el transcurso del video, emplear dicha curva sin ninguna modificación hará que el sistema pierda irremediablemente al objeto que pretende seguir. Esto ocurrirá porque el sistema estará intentando encontrar el objeto a seguir en el lugar equivocado. Asumiendo que los parámetros del modelo (A_1, A_2, B_0) se encuentran correctamente calculados y se cuenta con una cantidad de muestras tal que los cálculos estadísticos tengan sentido, es de esperar que el valor estimado de $X(\tilde{X}(t_k))$ no diste mucho de la posición y forma real del objeto³⁸ seguido. De todas formas la ubicación del sector de interés tampoco debe ser totalmente precisa, sino que se debe ubicar en el mismo sector que el objeto a seguir; de la ubicación exacta del objeto se encargará la observación utilizando los segmentos normales.

En el algoritmo de ajuste la matriz de covarianza $\tilde{P}(t_k)$ se emplea para fijar la longitud de los segmentos normales y en el mecanismo de puerta de validación para la eliminación de outliers. Tal como se vio en 4.3.2 cuando se mencionó la puerta de validación, cada vez

³⁷ En este punto se puede notar que la función de densidad *a priori* ayuda a la observación, en vez de simplemente corregirla una vez realizada.

³⁸ En el caso extremo, donde el sistema alcanza el estado de equilibrio (*steady state*), la estimación coincide exactamente con la observación.

que se omite un punto por considerarlo outlier, la matriz de incertidumbre S detiene su crecimiento. Aquí se puede ver cómo este mecanismo utiliza la matriz de covarianza calculada en el paso anterior, la cual juega un importante papel en el cálculo de la incertidumbre presente en el sistema. En la siguiente sección se mostrará cómo una mayor incertidumbre se traduce en que los valores obtenidos en la predicción tengan un menor peso en el resultado final de la iteración del algoritmo.

5.8.3 Asimilación

En este paso del algoritmo se realiza una asimilación entre la predicción y la medición. Es decir que se ajusta la estimación de acuerdo a lo que se acaba de observar, la cual sería una función típica de un regularizador. En la Figura 5-8 se puede ver un pseudo código de esta sección del algoritmo

$$\begin{aligned}
 1. K'(t_k) &= \tilde{P}'(t_k) [S(t_k) \tilde{P}(t_k) + I]^{-1} \\
 2. K(t_k) &= \tilde{P}(t_k) [S(t_k) \tilde{P}(t_k) + I]^{-1} \\
 3. P^*(t_k) &= \tilde{P}^*(t_k) - K'(t_k) S(t_k) \tilde{P}'(t_k) \\
 4. P'(t_k) &= \tilde{P}'(t_k) - K(t_k) S(t_k) \tilde{P}(t_k) \\
 5. P(t_k) &= \tilde{P}(t_k) - K(t_k) S(t_k) \tilde{P}(t_k) \\
 6. \hat{X}'(t_k) &= \tilde{X}'(t_k) + K'(t_k) Z(t_k) \\
 7. \hat{X}(t_k) &= \tilde{X}(t_k) + K(t_k) Z(t_k)
 \end{aligned}$$

Figura 5-8. Pasos correspondientes a la asimilación del filtro de Kalman

En esta sección del algoritmo, se emplean los resultados obtenidos por los pasos predictivos ($\tilde{P}(t_k)$, $\tilde{X}(t_k)$) y de medición ($S(t_k)$, $Z(t_k)$).

A continuación se detallará que ocurre en cada línea del pseudo código de la Figura 5-8:

1-3-4-6. En estas líneas se guarda el estado de las variables que intervinieron en esta iteración, para ser usadas en la próxima.

2. En esta línea se calcula la matriz de *Ganancia de Kalman*. Esta matriz permite asignar un peso a la observación obtenida en el paso correspondiente a la medición, de acuerdo a la incertidumbre presente en el sistema. Recuerdese que la matriz $S(t_k)$ es la indicadora de la incertidumbre en la observación; si las componentes de esta matriz son pequeñas, se estará en presencia de una mayor incertidumbre. Cuando el sistema encuentra incertidumbre, automáticamente sospecha que el objeto seguido no está respetando la dinámica predicha³⁹; es decir que no puede confiar en las predicciones. La *Ganancia de*

³⁹ Esto puede ocurrir porque el objeto a seguir comienza a trazar una trayectoria "errática", porque repentinamente acelera su movimiento o simplemente porque el algoritmo tiene problemas para su reconocimiento dentro de la imagen. Cuando la observación falla completamente, S vale 0 y

Kalman asegura el hecho de que la observación tendrá un mayor peso al haber más incertidumbre en el sistema.

En el pseudo código, se puede ver que cuando las componentes de la matriz $S(t_k)$ son pequeñas, las componentes de su inversa serán grandes, y por consiguiente las componentes de $K(t_k)$ serán mayores.

5. En esta parte del algoritmo, se ajusta el valor de la matriz de covarianza ($P(t_k)$) de acuerdo a la incertidumbre que se observó en la parte de medición. Se puede decir que la incertidumbre del sistema se halla manejada de manera cíclica por $P(t_k)$ y $S(t_k)$.

7. Esta parte del filtro de Kalman es la encargada de devolver el resultado final. Aquí se hace un ajuste al valor de transformación estimado, de acuerdo a la observación (representada por el vector de observación $Z(t_k)$). Si la incertidumbre en el sistema es alta, el valor de las componentes de la *Ganancia de Kalman* será mayor con lo cual $Z(t_k)$ tendrá un mayor peso. Si por el contrario la incertidumbre es baja, $K(t_k)$ será pequeña haciendo que lo observado tenga menor peso. En el caso extremo en que la observación falle totalmente, $Z(t_k)=0$ el algoritmo se quedará con la estimación.

aunque hay mucha incertidumbre al algoritmo no le queda otra opción que confiar en las predicciones.



Capítulo 6: Implementación y resultados

En este capítulo se mostrará donde se ubica el sistema implementado en el contexto de los sistemas de seguimiento.

Se mostrarán algunos ejemplos de corrida y se analizarán los resultados obtenidos. Luego de esto serán analizados los problemas que se encontraron al hacer funcionar el sistema.

Finalmente se comentará cual puede ser el posible trabajo a futuro con la intención de mejorar el sistema de seguimiento implementado y se expondrán las conclusiones finales del trabajo.

6.1 El sistema implementado en el contexto de los sistemas de seguimiento

Como se mencionó anteriormente, un esquema en el cual una spline o contorno se deforma para adaptarse al contorno de un objeto que se desea seguir, se denomina esquema de *Contornos Activos*. Si además existe un modelo asociado que permite facilitar el seguimiento del objeto (por ejemplo un modelo del tipo de movimiento realizado), entonces el esquema se denomina de *Contornos Dinámicos*. En el presente trabajo se emplea un esquema de *Contornos Dinámicos*.

En base a lo expuesto en la sección 1.3, el sistema de seguimiento implementado en este trabajo emplea *métodos basados en modelos*. Se emplea una *curva patrón deformable*, así también como un *espacio planar afin*. Por otro lado la predicción del movimiento se modela con *momentos auto-regresivos de segundo orden*, los cuales son empleados en el contexto del filtro de Kalman. Para extraer las propiedades a observar de la imagen, con el fin de descubrir el objeto a seguir, se emplea un *método basado en bordes*, compuesto por una *substracción del fondo*, seguida de un filtro simple. Por último se emplean *métodos basados en minimización de energía* para la obtención de la transformación que se aplicará a la curva patrón.

El modelo elegido: ventajas y desventajas

Se ha mencionado anteriormente que el espacio de formas empleado en el presente trabajo es el planar afin. Esta elección no es arbitraria, ya que, en general depende de cuánto se conoce a priori del movimiento (y forma) que tendrá el objeto a seguir. Por ejemplo, si el objeto a seguir sólo se traslada, no será necesario utilizar un espacio de formas tan amplio como el planar afin; con un espacio de formas que sólo contemple las traslaciones se puede ahorrar mucho tiempo de cálculos puesto que no se utilizarían

matrices de dimensiones mucho mayores que lo necesario⁴⁰. Además, el hecho de contar con una mayor cantidad de transformaciones hace que el sistema de seguimiento sea más sensible a los ruidos que puede haber en un cuadro de video. Por otro lado, si se restringe demasiado el espacio de transformaciones, sólo se obtendrá una vaga aproximación de la posición y/o forma del objeto en cada cuadro de la película. La Figura 6-1 muestra un ejemplo de un sistema de seguimiento que emplea un espacio de formas que contempla sólo traslaciones. Si el objeto se mueve de acuerdo a este espacio de formas, el sistema de seguimiento se muestra robusto.

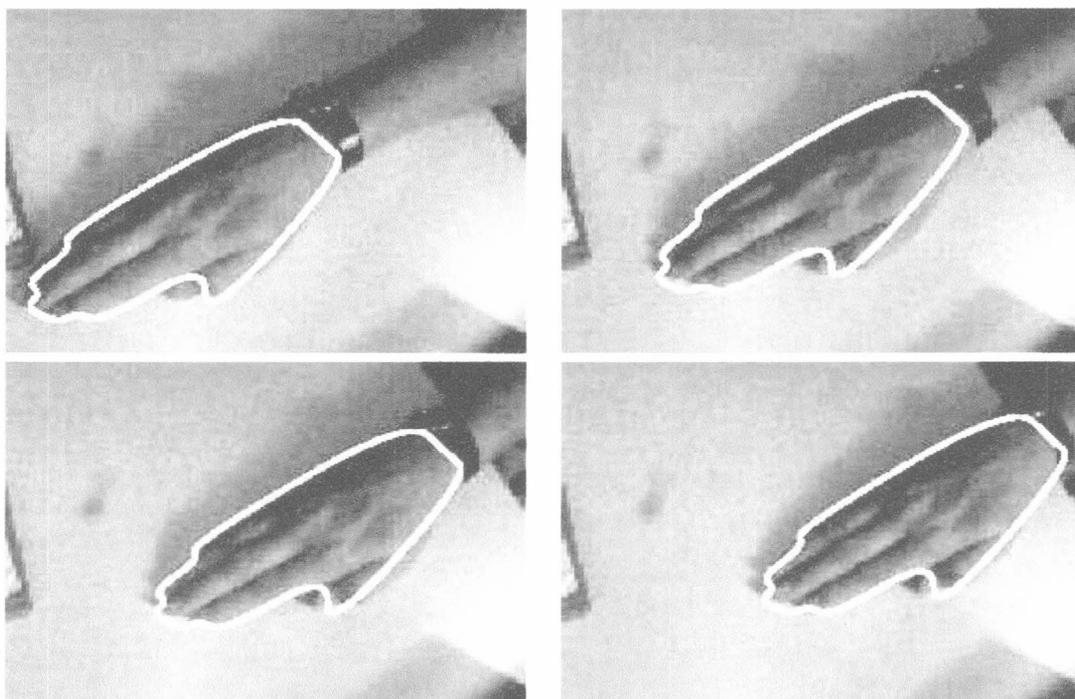


Figura 6-1. Sistema de seguimiento que emplea un espacio de formas donde sólo se contemplan traslaciones. *Cortesía de R. Curwen.*

Si el comportamiento del objeto a seguir presenta transformaciones no previstas por el espacio de formas, el sistema de seguimiento se vuelve deficiente. Esto puede verse en la Figura 6-2 donde el comportamiento del objeto presenta rotaciones y el espacio de formas sólo contempla traslaciones.

Debido a los pro y contras presentados, se debe buscar una solución de compromiso entre la cantidad de grados de libertad y la sensibilidad del sistema al ruido de fondo. Si se elige una mayor cantidad de grados de libertad, el seguimiento del objeto será más preciso ya que se permiten deformaciones más complejas de la curva patrón. Por otro lado, al tener más opciones de deformación es posible que el sistema se confunda con el ruido del fondo y elija una transformación inadecuada.

⁴⁰ Si se quiere saber con precisión qué tipo de movimiento realiza un objeto con el propósito de establecer el espacio de transformaciones necesario, lo más conveniente es emplear métodos creados para tal propósito tales como Análisis de Componentes Principales (PCA).

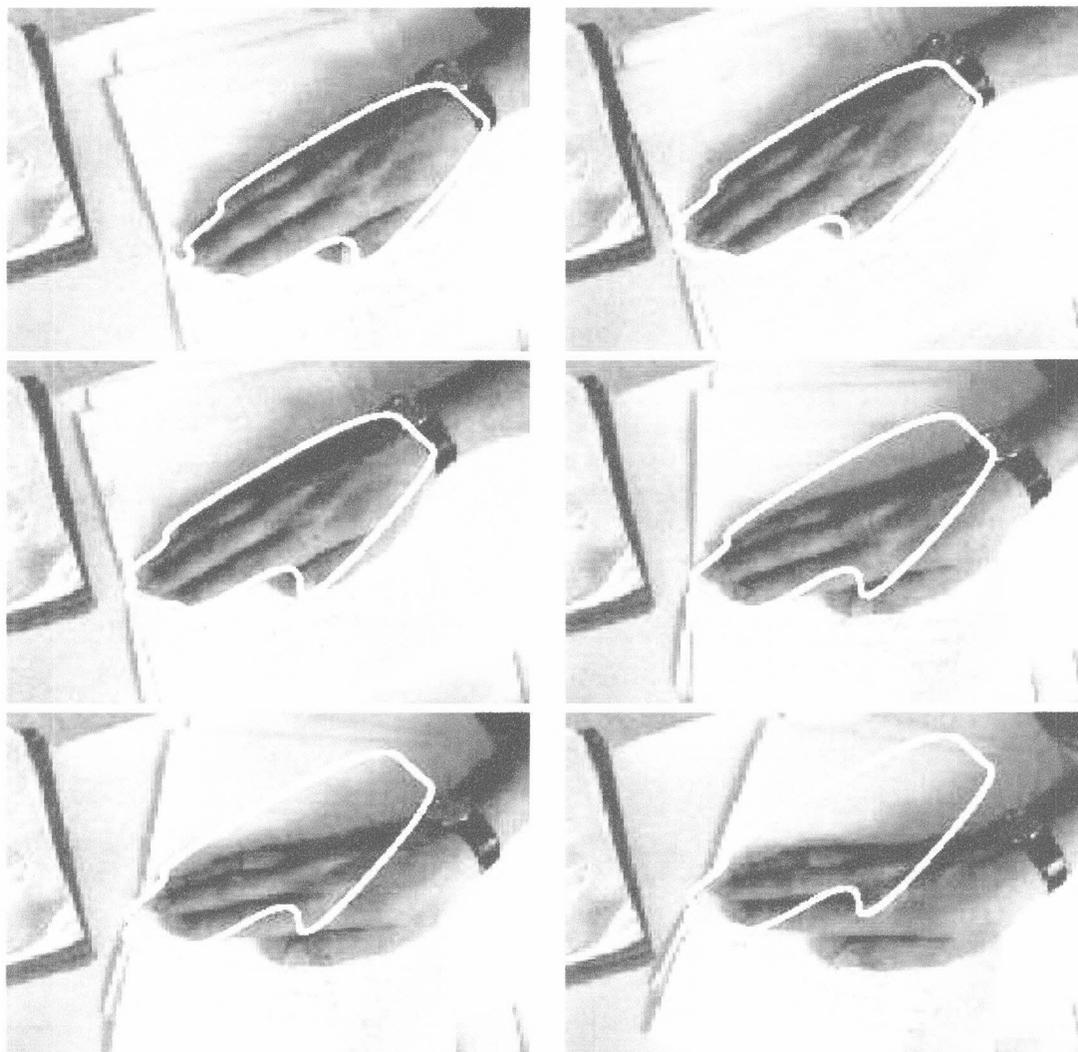


Figura 6-2. En este sistema de seguimiento se utiliza el mismo espacio de formas que en el ejemplo de la figura anterior pero el objeto a seguir realiza rotaciones que escapan al mismo. *Cortesía de R. Curwen.*

La elección de tener o no una curva patrón también es un factor importante a la hora de armar un sistema de seguimiento. Un sistema que no contenga una curva patrón rígida será muy sensible al ruido y, aunque se lograra controlarlo, permitirá una cantidad ilimitada de deformaciones sobre un mismo objeto. Un sistema que tenga una curva patrón, permitirá una cantidad de transformaciones limitada por la elección del espacio de formas elegido. Asimismo, la forma resultante siempre mantendrá la esencia de la curva patrón. Por otro lado, este tipo de sistemas será mucho más robusto en condiciones de ruido. En la Figura 6-3 se puede apreciar un sistema de seguimiento que no utiliza una curva patrón.

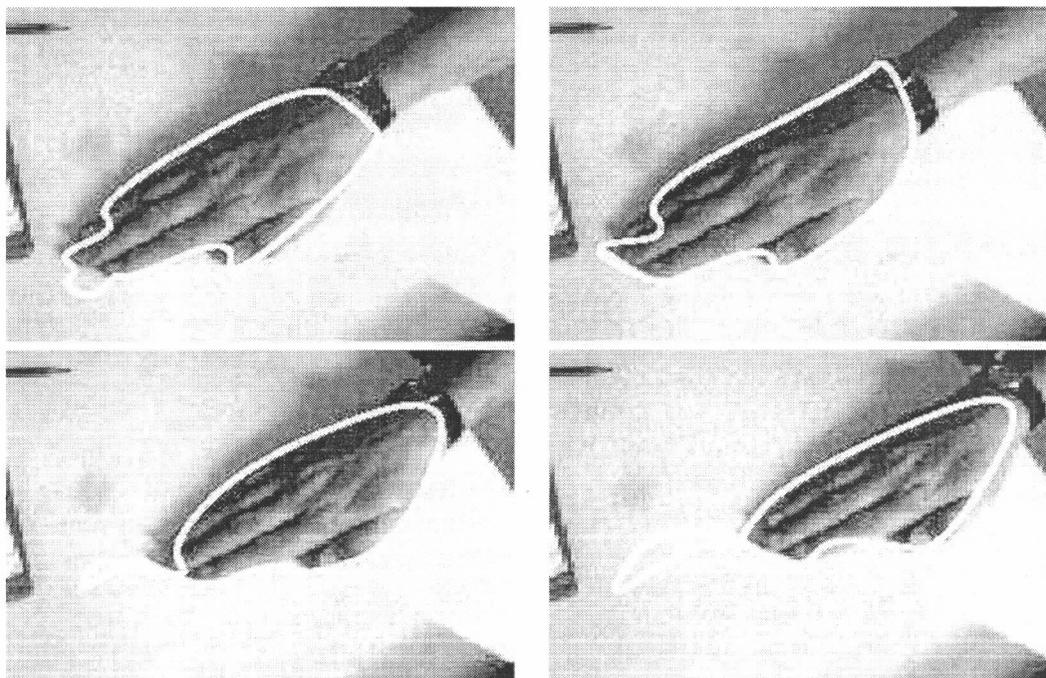


Figura 6-3. En esta figura se ve un sistema de seguimiento que no utiliza curva patrón. En el mismo se puede apreciar como el ruido afecta al resultado final. *Cortesía de R. Curwen.*

Por último, el costo computacional del algoritmo está íntimamente ligado a cuan complejas son las deformaciones posibles que puede presentar el modelo a la hora de seguir un objeto. Dado que las dimensiones de las estructuras de datos se hallan en función de las dimensiones del espacio de formas elegido, se puede ver que una mayor cantidad de grados de libertad se traduce en un procesamiento más lento. Esto es un punto muy importante a tener en cuenta, si es que se quiere modelar en tiempo real, como en el contexto del presente trabajo.

Teniendo en cuenta todas estas consideraciones, se ha optado por utilizar un espacio de transformaciones planar afín trabajando en conjunto con una curva patrón. Los grados de libertad del espacio planar afín permiten filtrar las transformaciones posibles. La curva patrón deformable tiene como objetivo mantener la esencia del objeto a seguir. Utilizar juntos ambos recursos hace que la curva resultante sea menos sensible a la oclusión y al ruido.

El modelo de curva patrón elegida para el desarrollo del presente trabajo es deformable en todos sus puntos de control por igual. La transformación obtenida como resultado del análisis de la imagen, se aplicará a todos los puntos de control por igual. Existen otros métodos que, aunque mantienen una curva patrón deformable, aplican transformaciones independientes en cada punto o conjunto de puntos de control. Esta diferencia no es menor y tiene incidencia en el tipo de contorno que la curva patrón puede aproximar. Por lo general, este mecanismo se emplea en casos donde se desea un mayor nivel de detalle. En la Figura 6-4 (foto de la derecha), al conjunto de puntos que se encuentra entre las piernas de la persona, se le ha aplicado una transformación adicional (un desplazamiento vertical) que permite obtener la deformación mostrada. Aunque este

procedimiento permite un mayor nivel de detalle, también su costo aumenta considerablemente [3].

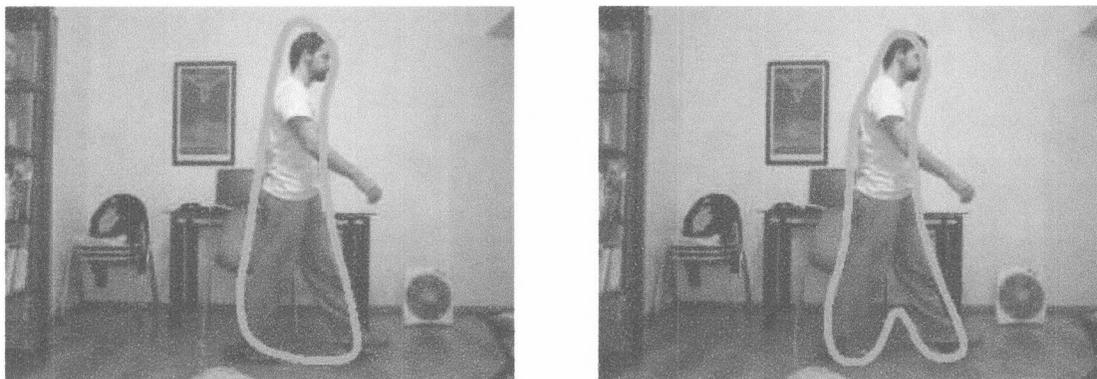


Figura 6-4. Dos formas de manejar curva patrón aplicados a la misma imagen. En la foto de la izquierda la transformación obtenida se ha aplicado a todos los puntos de la spline. En la foto de la derecha, se ha aplicado una transformación a todos los puntos de la curva y además se ha aplicado una transformación adicional en zonas donde se requiere un mayor detalle (en el ejemplo, las piernas de la persona).

Entre las desventajas del modelo elegido, se cuenta el manejo de los objetos tridimensionales como si fuesen planares. Para el sistema de seguimiento que se presenta, todos los objetos se encuentran paralelos al plano de la imagen. Esta decisión obedece principalmente a cuestiones de *performance*. Se pueden confeccionar espacios de formas que dispongan de transformaciones en el espacio, pero tal como se mencionó anteriormente el costo computacional se incrementa. Por otro lado, se pueden hacer correcciones a la transformaciones obtenidas mediante cálculos de pose y perspectiva, sin embargo, teniendo en cuenta el objeto de estudio (el cuerpo humano) no se consideró que se justifique el agregado de complejidad adicional.

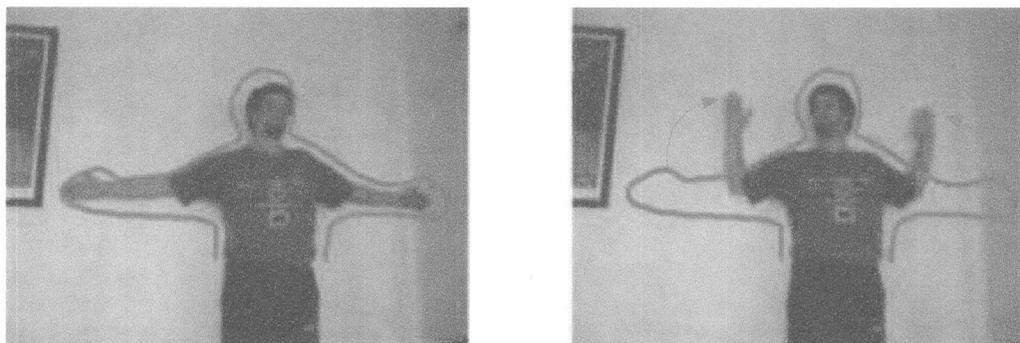


Figura 6-5. Una transformación del cuerpo humano que no puede ser representada por las transformaciones del espacio planar afín aplicadas por igual a todos los puntos de la curva patrón mostrada en la imagen.

Aplicar la misma transformación a todos los puntos de control de la curva patrón es una forma eficiente de deformar un contorno. Sin embargo, el cuerpo humano no presenta, por lo general, deformaciones que puedan ser modeladas por este mecanismo. En la Figura

6-5 se puede ver un contorno del cuerpo humano no representable por las transformaciones del espacio planar afín aplicadas a todos los puntos por igual.

Para solucionar el problema presentado en la Figura 6-5, se plantea el uso de varias curvas patrón para el seguimiento de un mismo objeto. Esto se verá más en detalle en la sección 6.4.

6.2 Implementación del algoritmo de ajuste

6.2.1 Muestreo por longitud de arco

Por más que el parámetro en una curva paramétrica crezca uniformemente, los puntos devueltos por la evaluación de la curva en dicho valor no se encuentran, por lo general, uniformemente distribuidos. Esto se puede ver en la Figura 6-6.

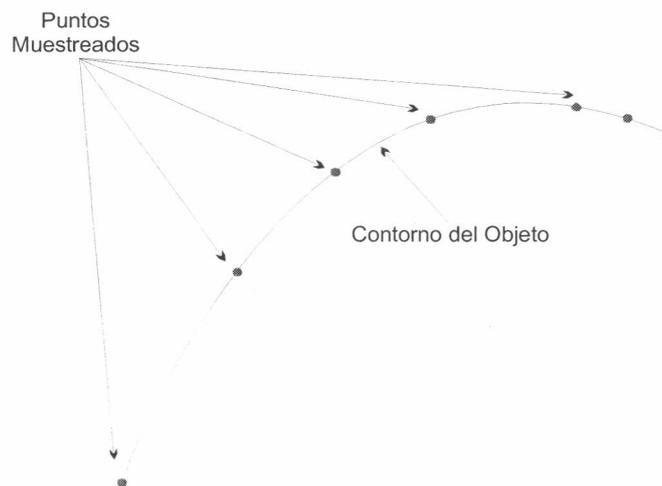


Figura 6-6. Los puntos devueltos por la evaluación de una curva paramétrica en el parámetro no se encuentran distribuidos uniformemente.

Por la forma en la que funciona el algoritmo de ajuste de la sección 3.2.2, la transformación resultado se forma a partir de la información recolectada por todos los puntos resultado del muestreo. El algoritmo pondera la información del muestreo y busca la transformación que mejor satisface a las muestras.

Lo primero que hace el muestreo por longitud de arco, es calcular la longitud de todo el contorno. Para hacer esto, lo usual es realizar una *discretización* del contorno en pequeños segmentos. Luego se suma la longitud de cada uno de los segmentos y se obtiene una aproximación de la longitud del contorno. Una vez hecho esto se divide la longitud calculada por la cantidad de puntos que serán muestreados, estableciendo de esta manera la distancia que debe haber entre cada muestra. Luego, en cada iteración del algoritmo de ajuste, en vez de incrementar el parámetro de la curva una cantidad fija, se busca en qué valor de dicho parámetro la evaluación de la curva devolverá un punto cuya distancia con el anterior es igual a la distancia prefijada. Esto hará que las muestras se encuentren igualmente espaciadas y que la transformación resultado del algoritmo de

ajuste se encuentre más balanceada satisfaciendo por igual a todos los sectores del contorno.

El muestreo por longitud de arco también se puede usar para seguir objetos empleando curvas cerradas cuando el objetivo es asegurar que la curva resultado satisfice por igual a todos los puntos de la curva patrón. Para realizar un seguimiento simple, las curvas cerradas no requieren de un anclaje⁴¹, ya que, por la forma en que contornean al objeto, es más difícil que lo pierdan por deslizamiento [13] como ocurre con las curvas abiertas.

Si existe un conjunto de muestras concentrado en un determinado sector de la curva, el algoritmo de ajuste hará que este último tenga más peso que un sector donde las muestras se hallen más dispersas. De esta forma la transformación resultado tenderá a satisfacer más a ciertos sectores del contorno.

A veces, es útil dar más peso a ciertos sectores del contorno, haciendo que la transformación resultado satisfaga más a algún sector específico. Como resultado inmediato de esto, donde haya en el sector una mayor concentración de puntos, el seguimiento se realizará de una manera más precisa⁴². En el presente trabajo, este mecanismo se emplea cuando se considera el seguimiento de objetos mediante una curva patrón que necesite estar *anclada* al objeto que se encuentra siguiendo el sistema.

6.2.2 La curva característica: una entidad teórica

La curva característica representa el contorno del objeto que se desea seguir. Sin embargo, se debe tener en cuenta que su existencia es una consideración teórica que se realiza con el fin de dar sustento a la teoría del algoritmo de ajuste. En la práctica, no importa la evaluación del parámetro en dicha curva, ya que la comparación se realiza entre $r(s)$ y el producto del muestreo de los segmentos normales en el valor del parámetro. La muestra obtenida por el segmento normal en la posición del parámetro, se considera perteneciente a la curva característica, razón por la cual el algoritmo es correcto.

Siendo que no se emplea la curva característica, no se requiere de su costosa construcción.

6.2.3 Optimización de la substracción del fondo

El uso de la substracción del fondo para la detección del objeto a seguir es costoso computacionalmente. Aunque en el presente trabajo se emplea la técnica de substracción del fondo, su aplicación se restringe únicamente al camino trazado por los segmentos

⁴¹ Más adelante será explicado en que consiste el anclaje de una curva patrón. Mientras tanto, se puede asumir que un anclaje es un punto donde la curva patrón se aferra al objeto que se encuentra siguiendo el sistema.

⁴² Se puede querer seguir con mayor precisión, por ejemplo, cierta parte del cuerpo de una persona y no otra.

normales. Al no hacerse una substracción completa del fondo y por lo tanto no considerando todos los píxeles de la imagen, su costo se reduce de manera considerable.

Lo primero que se hace, es tomar una imagen del fondo de la escena en el cual no se encuentre el objeto a seguir. Luego, cada píxel por el que pasa el segmento normal se compara contra el fondo; si su color es similar al color presente en la misma posición en la imagen del fondo⁴³, entonces el píxel en cuestión se considera de color *negro*. Los píxeles *negros*, no son tenidos en cuenta por el algoritmo de muestreo de normales ya que se consideran parte del fondo. De esta forma, el algoritmo avanzará por el segmento normal mientras los píxeles por los que pase sean *negros*, considerando que se ha llegado al contorno del objeto cuando se encuentra un píxel de distinto color⁴⁴. En la Figura 6-7 se puede ver un esquema del código empleado para la substracción del fondo.

```
Mientras pixel_corriente.color = negro  $\wedge$  no_termine_normal
...
  Si pixel_corriente.color  $\neq$  fondo(pixel_corriente.x, pixel_corriente.y).color  $\pm$   $\xi$ 
  {
    Indicar muestra encontrada
  }
  Sino
  {
    pixel_corriente.color = negro
  }
  Buscar pixel_siguiente
Fin Mientras
```

Figura 6-7. Esquema del fragmento del código que considera la substracción del fondo para encontrar el objeto a seguir.

En esta implementación, se puede observar que el filtro utilizado se limita a indicar si el píxel analizado encuentra o no una muestra. La razón por la cual se utiliza el valor de un único píxel se basa en que utilizar varios puntos puede conducir a encontrar muestras que no son las deseadas. Por ejemplo, cuando un segmento normal comienza dentro del objeto a seguir y termina fuera de éste, se encontrará una muestra al salir del objeto y no es lo que se desea. En la práctica, este filtro simple, acompañado del modelo de estimación seguimiento, demostró tener un buen comportamiento.

6.3 Implementación del filtro de Kalman

6.3.1 Recuperación en presencia de incertidumbre

Un objeto se encuentra en estado de *lock* cuando la cantidad de *outliers* que se detectan en la iteración se halla bajo cierto umbral. Si dicha cantidad se vuelve importante, será

⁴³ Con una tolerancia especificada.

⁴⁴ Indicando la existencia de una muestra en dicha posición.

necesario tomar medidas para que el sistema no se desestabilice, perdiendo al objeto. En la presente implementación se ha optado por volver la matriz de covarianza a su estado inicial (donde sus componentes son mayores). De esta forma el algoritmo puede llegar a recuperarse si por alguna razón se pierde al objeto⁴⁵. Una vez reinicializada la matriz de covarianza, el algoritmo comienza nuevamente un ciclo en el cual llegará a tener una mayor certidumbre y volverá a *lockear* al objeto. Este mecanismo de búsqueda de equilibrio constante funciona como una ayuda adicional que se le da al algoritmo en momentos de muy poca certidumbre, evitando, mientras sea posible, que falle completamente la observación. Un ejemplo del funcionamiento del mecanismo comentado se puede ver en la Figura 6-8.

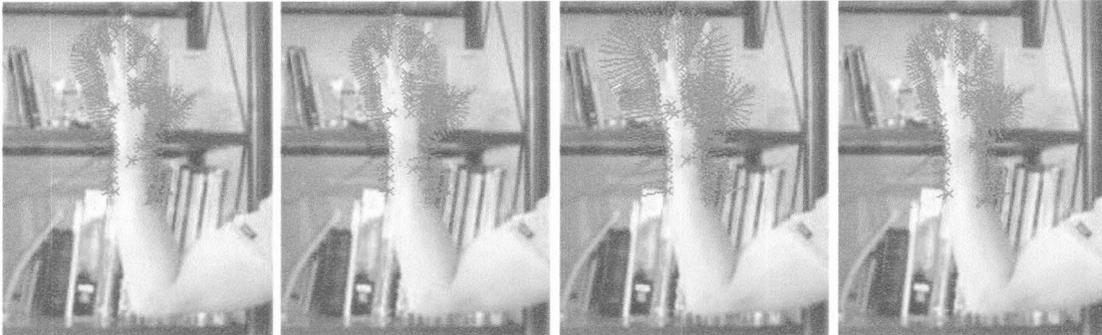


Figura 6-8. Un fragmento de secuencia de video donde se puede observar que en el tercer cuadro las normales son reseteadas como consecuencia de una pérdida de puntos. Se puede ver que la recuperación se realiza inmediatamente. Esto se debe principalmente a que el video tiene aprendidos parámetros dinámicos de movimiento (esto se verá más adelante en la sección 6.5.3).

En la jerga del seguimiento, esta reinicialización de la matriz de covarianza es equivalente a lo que se conoce como *entrada al estado open loop*⁴⁶.

Finalmente, se puede considerar una matriz de covarianza de *open loop* distinta para el sistema con y sin modelo de estimación de movimiento. Esto se puede hacer, debido a que un sistema que cuenta con un modelo de estimación de movimiento, suele tener una mayor certidumbre (en la sección 6.5.3 se verá un ejemplo de esto). Si se emplea la misma matriz de covarianza que se empleaba antes de aprender los parámetros dinámicos, los segmentos normales resultantes serán mucho más largos de lo necesario para encontrar la mayoría de los puntos del contorno del objeto. Esto hará que el sistema trabaje de más, recorriendo normales innecesariamente. El ahorro que se puede lograr empleando una matriz de covarianza con componentes más pequeñas en un sistema con estimación de movimiento, se traduce en un procesamiento más rápido. Sin embargo,

⁴⁵ Cabe recordar que una covarianza con componentes más grandes, se traduce entre otras cosas en una mayor longitud de los segmentos normales. Si estos son mayores, la probabilidad de encontrar al objeto también aumentará.

⁴⁶ Cuando el sistema está estable o se mantiene *lockeado* al objeto, se dice que está en estado *close loop*.

dicho ahorro tendrá lugar sólo cuando el sistema entra al *open loop*⁴⁷, lo cual se espera que no ocurra con frecuencia cuando se emplea un modelo de estimación de movimiento.

6.3.2 Simulando un estado de equilibrio

En 6.3.1 se comentó que cuando la cantidad de puntos encontrados con los segmentos normales se encuentra bajo un cierto umbral, es necesario tomar medidas para evitar que se desestabilice el sistema. En el caso opuesto, donde la cantidad de puntos encontrados se encuentra *sobre* un cierto umbral, el algoritmo del filtro de Kalman puede ser modificado para obtener un funcionamiento más rápido.

Cuando en el sistema existen pocos outliers, se considera que la incertidumbre es baja; en este caso, el modelo de movimiento funcionará mejor, pudiendo incluso prescindir de la observación por un corto período de tiempo. La idea es que llegado un punto donde la cantidad de outliers se encuentre bajo cierto umbral, se pase a un estado denominado *equilibrio dinámico*. En este estado, si bien el sistema no se encuentra en estado de equilibrio en lo que respecta a las transformaciones que sufre el objeto que se está siguiendo, es la incertidumbre del sistema, cuyo valor es bajo, la que está en equilibrio. En el estado de *equilibrio dinámico* se trata de aprovechar al máximo el modelo de movimiento aprendido y no se realizan observaciones. Si el sistema presenta una incertidumbre tan baja como atestigua el algoritmo de ajuste, esto indicará que el objeto que se encuentra siguiendo el sistema se mueve de acuerdo a la dinámica que presenta el modelo de movimiento. Como resultado, durante el estado de *equilibrio dinámico* el sistema puede funcionar mucho más rápido ya que no se realizan los costosos cálculos involucrados en la observación.

Al igual que en el estado de equilibrio estacionario (*steady state*), el estado de *equilibrio dinámico* debe usarse con discreción. Dado que no se realizan observaciones, si el objeto cambia bruscamente su patrón de movimiento, el sistema lo perderá irremediamente. En la presente implementación se alterna la observación con el estado de *equilibrio dinámico*, de forma de evitar que se pierda el objeto a seguir. Esto se puede ver reflejado en el organigrama de la Figura 6-9.

En la Figura 6-9 se puede ver cómo se pasa a un estado de equilibrio dinámico cuando la cantidad de outliers del sistema se encuentra bajo cierto umbral. Dentro del equilibrio dinámico, el funcionamiento es similar al del filtro de Kalman convencional, solo que no se realizan observaciones. El resultado final de la iteración se calcula sólo con valores devueltos por los momentos auto regresivos.

Finalmente, una vez que se han realizado n iteraciones del filtro de Kalman sin observación, se vuelve al funcionamiento clásico. El valor de n dependerá del tipo de objeto a seguir; si éste se mueve lentamente, el valor de n puede ser mayor, ya que el

⁴⁷ Como se vio en el ejemplo de la Figura 6-8, la convergencia de la certidumbre luego de salir del *open loop* es casi inmediata, por lo que casi no existen longitudes intermedias entre el segmento normal del *open loop* y el segmento normal promedio que maneja el sistema. De esta manera el ahorro prácticamente se hace efectivo sólo en el cuadro en el cual el sistema entra al *open loop* (el tercero en la figura).

sistema se recuperará más fácilmente si el objeto abandona la dinámica esperada, es decir que, se puede mantener durante un mayor tiempo el *equilibrio dinámico*.

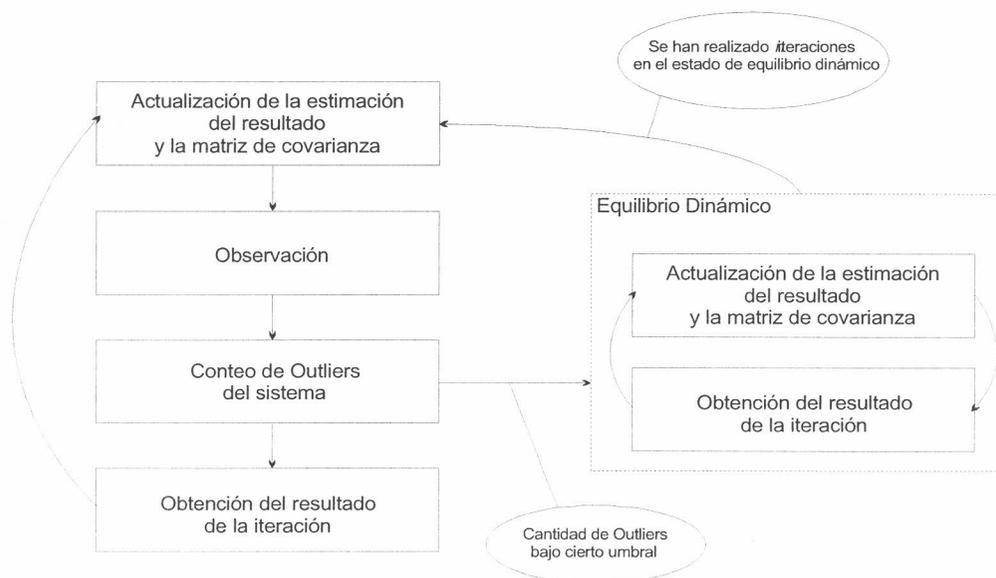


Figura 6-9. Organigrama del uso del *equilibrio dinámico* en el filtro de Kalman. Cuando la cantidad de outliers se encuentra bajo cierto umbral, el sistema pasa al estado de equilibrio dinámico. Allí el funcionamiento del filtro de Kalman es equivalente al caso normal, sólo que no se realizan observaciones. Una vez que se han realizado tantas iteraciones en el estado de equilibrio dinámico como se ha configurado, se vuelve al funcionamiento normal del sistema.

Por otro lado, si se espera que el objeto haga movimientos bruscos o rápidos, lo mejor es considerar un valor de n pequeño de manera que solo se suspenda la observación en unos pocos cuadros. En este último caso, es necesario utilizar un valor pequeño de n , ya que si el sistema perdiera al objeto (y no se enterara debido a que no se encuentra observando), la recuperación sería mucho más costosa y a veces imposible.

6.4 Trabajo con Varias Curvas

Para modelar transformaciones que se escapan al espacio de formas planar afín se han hecho algunas modificaciones al sistema, de manera que funcione con múltiples curvas. Estas curvas son totalmente independientes, teniendo cada una sus propios parámetros dinámicos (prefijados o aprendidos por el algoritmo de aprendizaje).

Manejar distintos conjuntos de parámetros dinámicos para las distintas curvas permite que cada una de ellas pueda seguir una parte del mismo objeto y en conjunto poder modelar transformaciones no planares afines. De esta manera, transformaciones como las realizadas en la Figura 6-5, pueden ser modeladas como se muestra en el ejemplo de la Figura 6-10.



Figura 6-10. Secuencia de video que muestra el seguimiento de transformaciones del cuerpo humano que escapan al espacio planar afín y que son resueltas mediante múltiples curvas funcionando de manera independiente.

A pesar de agregar flexibilidad al sistema haciéndolo más resistente a transformaciones no planares afines, existen problemas que se pueden presentar a la hora de trabajar con varias curvas. El más importante de los problemas aparece cuando los fragmentos de objeto seguidos por varias curvas, se ocluyen mutuamente. En este caso, las curvas que seguían a los fragmentos ocluidos no podrán discernir cual era el objeto que se encontraban contorneando. En ese caso se requiere hacer un aprendizaje más intensivo de forma tal que las curvas sean más resistentes a las oclusiones (ver sección 6.5.2).

6.5 Usando el sistema de Seguimiento

En la presente sección se mostrará el sistema de seguimiento implementado, corriendo sobre algunos videos de ejemplo. Los videos empleados fueron obtenidos por una cámara *Canon S30* a 15 cuadros por segundo. Asimismo el seguimiento fue realizado con un procesador *Intel Pentium 4* de 2Ghz y 256 megabytes de memoria RAM.

Todos los ejemplos mostrados, representan secuencias de video cuyo orden ha sido establecido de izquierda a derecha y de arriba a abajo.

6.5.1 Algunos ejemplos

En este punto se mostrarán y analizarán brevemente algunas de las secuencias de video usadas como ejemplo, luego de su procesamiento con el sistema de seguimiento implementado en el presente trabajo.

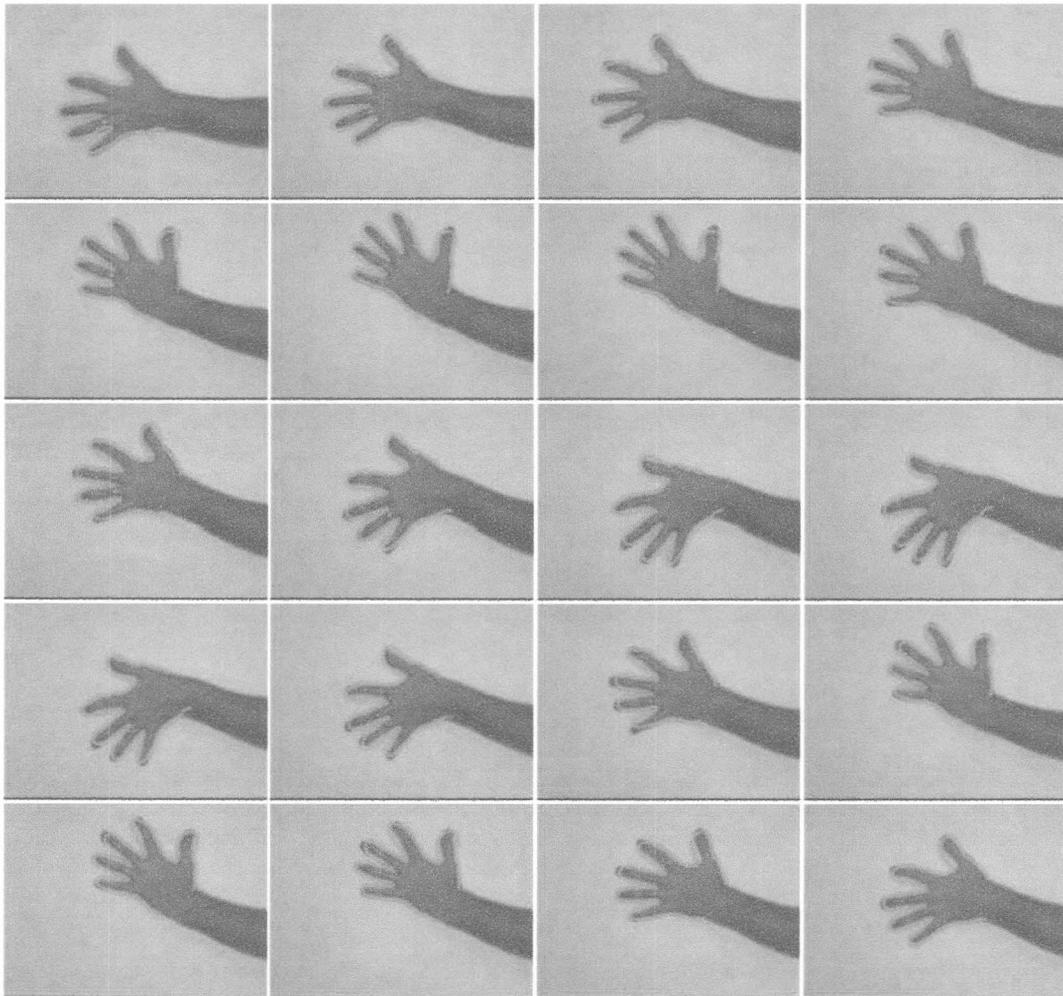


Figura 6-11. Secuencia de video que muestra la rotación de una mano en el plano paralelo a la imagen. La duración es de aproximadamente 20 segundos.



Figura 6-12. Secuencia de video que muestra un brazo realizando rotaciones en el plano paralelo a la imagen. La duración del video es de aproximadamente 30 segundos.

En la Figura 6-11 se puede observar la rotación de una mano en el plano paralelo a la imagen. El fondo es simple y fácil de manejar por los métodos de substracción del fondo. Asimismo la curva patrón se halla bien encajada en el objeto.

El video se confeccionó luego de una iteración de aprendizaje. Sin embargo, dado que los movimientos son lentos y el fondo es simple, el seguimiento no mejoró de manera considerable al agregar un modelo de estimación de movimiento.

El fondo simple, una buena curva patrón y movimientos típicos del espacio de formas considerado, hacen que el seguimiento sea realizado de manera satisfactoria. De hecho, este video entraría en la categoría de *videos de aprendizaje*, es decir, aquellos empleados para obtener los parámetros del modelo de estimación de movimiento.

La secuencia de video de la Figura 6-12 también podría considerarse perteneciente a la categoría *videos de aprendizaje*. Sin embargo, aunque el fondo se halla relativamente *limpio*, se puede ver que las transformaciones necesarias para seguir con precisión el brazo en movimiento no son planares afines⁴⁸. Esto se nota sobre todo en los últimos 4 cuadros de la Figura 6-12, donde el brazo realiza un movimiento que lo saca del plano de la imagen. Si bien el sistema de seguimiento no se encuentra preparado para realizar transformaciones que no pertenezcan al espacio de formas planar afín, se puede ver en la Figura 6-12 que el sistema logra seguir el brazo de manera *aproximada* sin desestabilizarse.

Más adelante se verá como el video de la Figura 6-11 muestra ser eficiente para realizar sobre él, un aprendizaje de parámetros dinámicos.

En la Figura 6-13 se muestra al sistema de seguimiento trabajando con una persona que realiza una caminata paralela al plano de la imagen. En este ejemplo, el objeto que se desea seguir se mueve con una mayor velocidad que en el video anterior, haciendo que el sistema de seguimiento tenga un poco más de problemas para no perderlo. El sistema fue entrenado una serie de veces hasta lograr los resultados que se muestran en la Figura 6-13. En el video solo se muestran traslaciones, siendo el mayor problema para el sistema de seguimiento el evitar que la curva patrón se *desprenda* del objeto a seguir. Esto se puede observar si se mira el extremo izquierdo de la curva resultado, la cual se ve más *despegada* del objeto en algunos cuadros del video. Para evitar esta clase de problemas, se debe recurrir a curvas patrón que tengan un buen *anclaje* en el objeto que se encuentran siguiendo.

⁴⁸ En estos casos lo que se puede hacer es emplear la secuencia para aprendizaje hasta el punto donde el objeto comienza a realizar movimientos que no pueden ser representados por el espacio de formas planar afín. Esto se puede hacer porque ya se tiene una cantidad suficiente de información respecto del movimiento que realiza el brazo. El movimiento es cíclico y se ha estudiado por lo menos un ciclo del mismo.

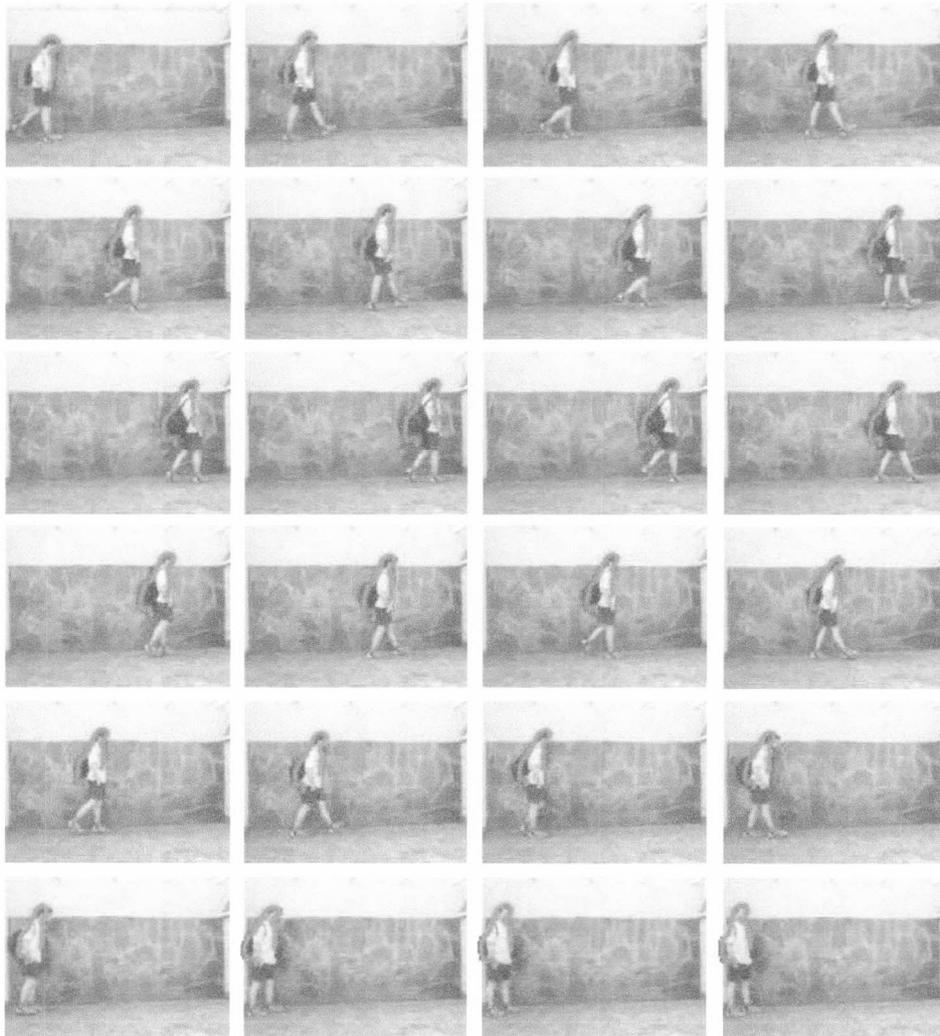


Figura 6-13. Secuencia de video que muestra al sistema de seguimiento siguiendo a una persona que realiza una caminata paralela al plano de la imagen. La duración del video es de aproximadamente 10 segundos.

La Figura 6-14 muestra al sistema de seguimiento implementado, funcionando con una curva que abarca toda la silueta de la persona. Si se observa el detalle de las piernas, se puede ver que el sistema de seguimiento tiene problemas para seguir el movimiento. Esto ocurre debido a que las deformaciones que sufre la curva patrón ocurren en el espacio planar afín y el movimiento que presentan las piernas de una persona caminando escapa a las transformaciones planares afines⁴⁹.

⁴⁹ Si se quiere seguir ambas piernas con la misma curva, no existirá una transformación en el espacio de formas planar afín aplicable a dicha curva tal que modele el movimiento correctamente. Más adelante se planteará un esquema que presenta una solución a este problema.

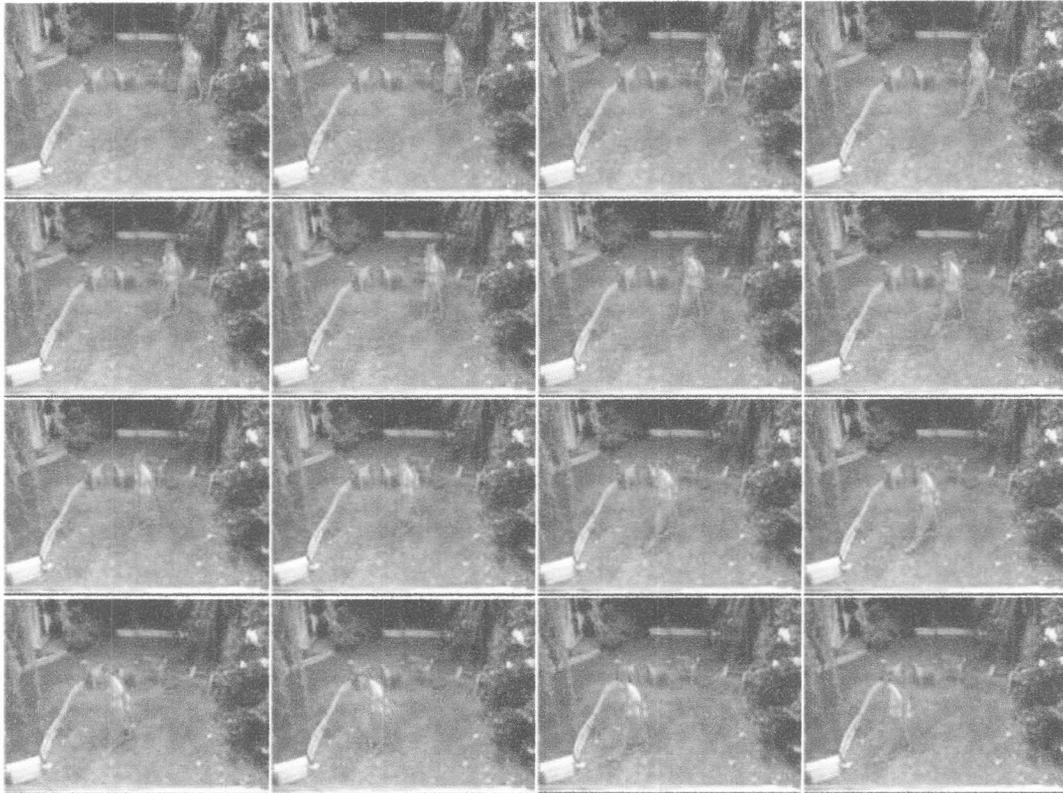


Figura 6-14. Secuencia de video que muestra una caminata, en un plano oblicuo al de la imagen. La duración de la secuencia de video es de aproximadamente 4 segundos.

6.5.2 Ventajas del Aprendizaje

Aprender los parámetros dinámicos de un proceso auto regresivo permite construir un sistema de seguimiento robusto, especialmente ajustado al tipo de movimiento que se desea seguir. A continuación se mostrará que el aprendizaje de dichos parámetros lleva a un mejor seguimiento, fortaleciendo al sistema de seguimiento y haciéndolo más resistente a las eventuales oclusiones que puede sufrir el objeto seguido.

Mejor Seguimiento

Puede darse el caso en que un sistema de seguimiento no puede seguir un objeto debido a que el mismo se confunde con el fondo de la escena donde se encuentra⁵⁰. En la Figura 6-15 se puede observar como el sistema de seguimiento pierde al objeto que se desea

⁵⁰ Se asume que esta confusión no puede ser resuelta manipulando la tolerancia de la substracción del fondo.

seguir (la persona caminando hacia la izquierda) ya que no utiliza parámetros dinámicos aprendidos previamente⁵¹.



Figura 6-15. Seguimiento sin aprendizaje de los parámetros dinámicos. En la imagen se puede ver que a partir de cierto cuadro el sistema se confunde con el entorno y la curva pierde al objeto seguido. El sistema de seguimiento no puede recuperarse de dicha pérdida con los parámetros establecidos.

En la Figura 6-17 se observa como el mismo sistema de seguimiento puede seguir al objeto ya que utiliza parámetros dinámicos aprendidos con anterioridad (obtenidos con la secuencia de video de la Figura 6-16).

⁵¹ No utilizar un sistema de estimación de movimiento no implica que el sistema de seguimiento no pueda seguir a un objeto; sin embargo, si el objeto se encuentra en una escena con mucho ruido, el sistema de estimación de movimiento hace al sistema de seguimiento más robusto.



Figura 6-16. Secuencia de video empleada para la obtención de parámetros dinámicos.



Figura 6-17. Seguimiento con aprendizaje de los parámetros dinámicos. La secuencia de video es la misma que la de la Figura 6-15. Sin embargo, como el sistema utiliza los parámetros dinámicos aprendidos con la secuencia de video de la Figura 6-16, no se confunde con el entorno y puede mantener el seguimiento de la persona.

La razón por la cual un sistema de seguimiento funciona y el otro no, radica fundamentalmente en el uso de los parámetros dinámicos. El sistema que utiliza los parámetros aprendidos puede predecir con gran exactitud donde se encontrará el objeto a seguir, ubicar al sector de interés con una mayor precisión y realizar mejores correcciones a la observación.

En la Figura 6-18 y la Figura 6-19 se puede ver otro ejemplo que muestra las ventajas de haber realizado un aprendizaje previamente. En la Figura 6-18 el sistema no puede seguir al objeto debido a que este se mueve con demasiada velocidad. En la Figura 6-19, donde se cuenta con un aprendizaje realizado, se puede observar que a pesar de que al sistema le cuesta realizar el seguimiento, el objeto no se pierde.



Figura 6-18. Secuencia de video que muestra un brazo que se mueve a gran velocidad sobre un fondo complejo. Al no haber un aprendizaje previo, el sistema pierde al objeto.

Un sector de interés bien ubicado⁵² hará que se pierda una menor cantidad de muestras, con lo cual el seguimiento será mejor. Además, si el sistema tiene parámetros aprendidos, la incertidumbre será menor (ver sección 6.5.3), con lo cual de acuerdo a Figura 5-8: Paso 7 el sistema le dará un menor peso a la observación. Por otro lado, si el sistema le da mucha importancia a la observación⁵³ un entorno con ruido hará que el seguimiento no se realice correctamente ya que el sistema se confundirá con el entorno.

⁵² La ubicación del sector implica su traslación y deformación.

⁵³ Hecho que sucede cuando la incertidumbre es alta, como en este caso.



Figura 6-19. Seguimiento realizado con un modelo de estimación de movimiento aprendido a partir de la secuencia de video de la Figura 6-12. Esta secuencia de video es mucho más rápida que la empleada para el aprendizaje y gracias al modelo de estimación de movimiento el sistema no pierde al objeto.

Manejando las oclusiones

Otra ventaja que se obtiene con el aprendizaje de los parámetros dinámicos, se relaciona con la oclusión de los objetos. Si un sistema de seguimiento no cuenta con parámetros aprendidos, es muy difícil que pueda seguir a un objeto que se ocluye. En la Figura 6-20 se puede ver un ejemplo de esto.



Figura 6-20. Seguimiento sin aprendizaje de los parámetros dinámicos: En esta imagen se ve como el sistema de seguimiento confunde al objeto que desea seguir con la persona que está pasando por delante de la cámara. El sistema de seguimiento no puede recuperarse ante la oclusión ocurrida.

En la Figura 6-21 se utiliza un sistema de seguimiento en donde se han aprendido los parámetros dinámicos del movimiento deseado (de acuerdo a la secuencia de video de la Figura 6-16). Se puede observar como una vez terminada la oclusión, el sistema recupera al objeto que se hallaba siguiendo.



Figura 6-21. Seguimiento con aprendizaje de los parámetros dinámicos: Si bien el sistema de seguimiento confunde al objeto que desea seguir con la persona que está pasando por delante de la cámara, el mismo utiliza los parámetros aprendidos para poder recuperar al objeto que estaba siguiendo.

6.5.3 Evolución de la Incertidumbre

En esta parte del trabajo se analizará de que manera la incertidumbre fluctúa en el sistema de seguimiento, dependiendo tanto de la observación que se encuentra realizando el sistema como del modelo aprendido.



Figura 6-22. Secuencia de video de aprendizaje, que muestra un individuo caminando oblicuamente al plano de la imagen. La duración del video es de aproximadamente 10 segundos.

Considérese el video de la Figura 6-22, donde se puede ver una persona caminando oblicuamente al plano de la imagen. Esa caminata oblicua al plano de la imagen, representa una transformación que no se encuentra contenida en el espacio de formas planar afin: la perspectiva. Sin embargo, y dado que el sujeto se encuentra a una relativa distancia de la cámara, el sistema resuelve el seguimiento de manera satisfactoria presentando una pequeña desestabilidad conforme el objeto se va acercando⁵⁴.

Si de este mismo video se extraen los parámetros dinámicos del modelo (es decir que se emplea para aprender), y se vuelve a correr el sistema de seguimiento, el resultado es el que se observa en la Figura 6-23. En dicha figura, la curva se acomoda mejor sobre el objeto y el procesamiento del video consume ligeramente una menor cantidad de

⁵⁴ En el infinito, la proyección de perspectiva tiende a la proyección ortográfica. Por otro lado, la perspectiva del objeto se acentúa conforme este se acerca a la cámara.

tiempo⁵⁵. La pequeña reducción en el tiempo de procesamiento, se debe a la disminución de la incertidumbre presente en el sistema, conseguida gracias al modelo de estimación de movimiento configurado.

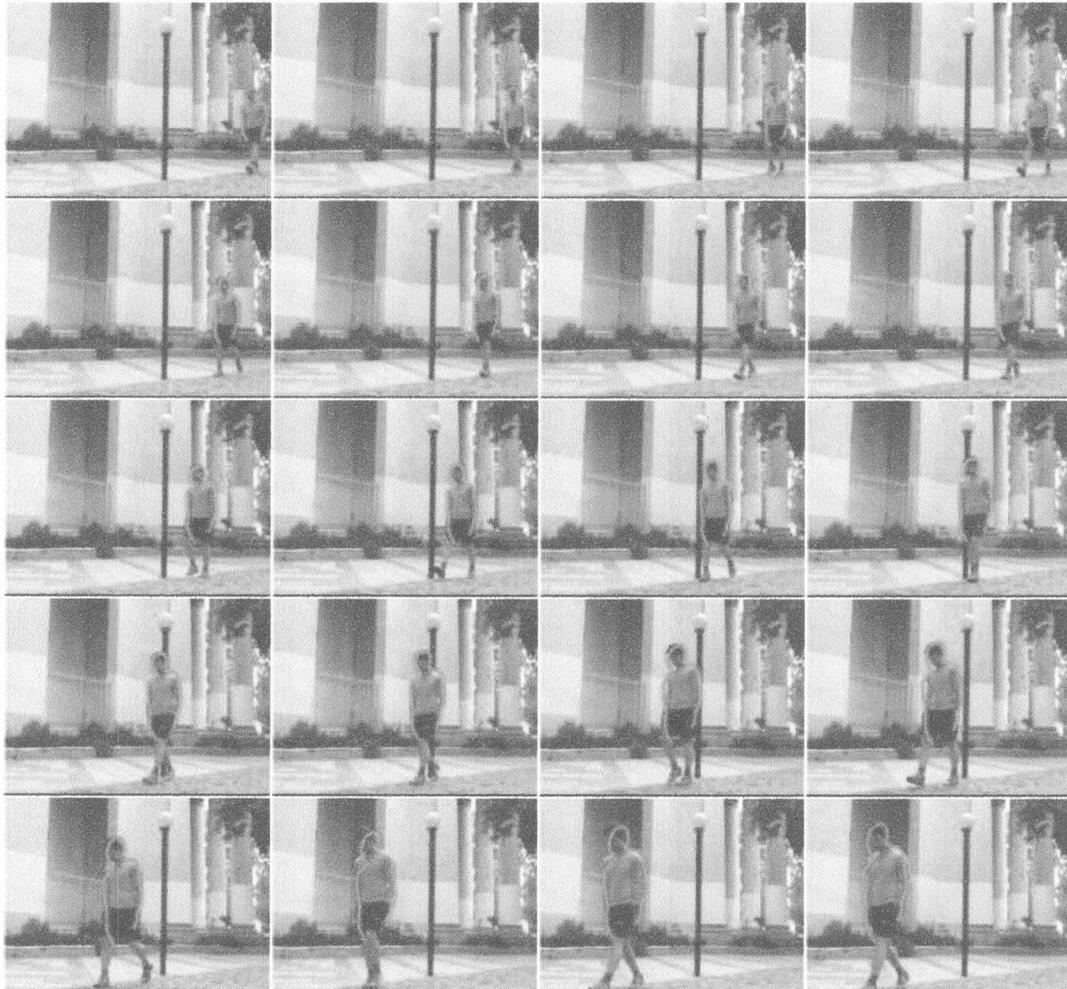


Figura 6-23. Secuencia de video, que muestra un individuo caminando oblicuamente al plano de la imagen. Aquí se han empleado los parámetros de dinámicos obtenidos por el procesamiento que dio como resultado la figura anterior.

El modelo de estimación de movimiento configurado, hace que el sistema pueda ubicar mejor el sector de interés. Una mejor ubicación de dicho sector, se reflejará en una menor cantidad de puntos perdidos y por consiguiente una menor cantidad de entradas al *open loop* (ver sección 6.3.1). Esto se puede ver en la Figura 6-24.

⁵⁵ La diferencia es de aproximadamente 10 centésimas de segundo para este caso en particular. Si se considera el tiempo de procesamiento de video estándar (24 cuadros por segundo), esto equivale a procesar 2.4 cuadros más por cada segundo.

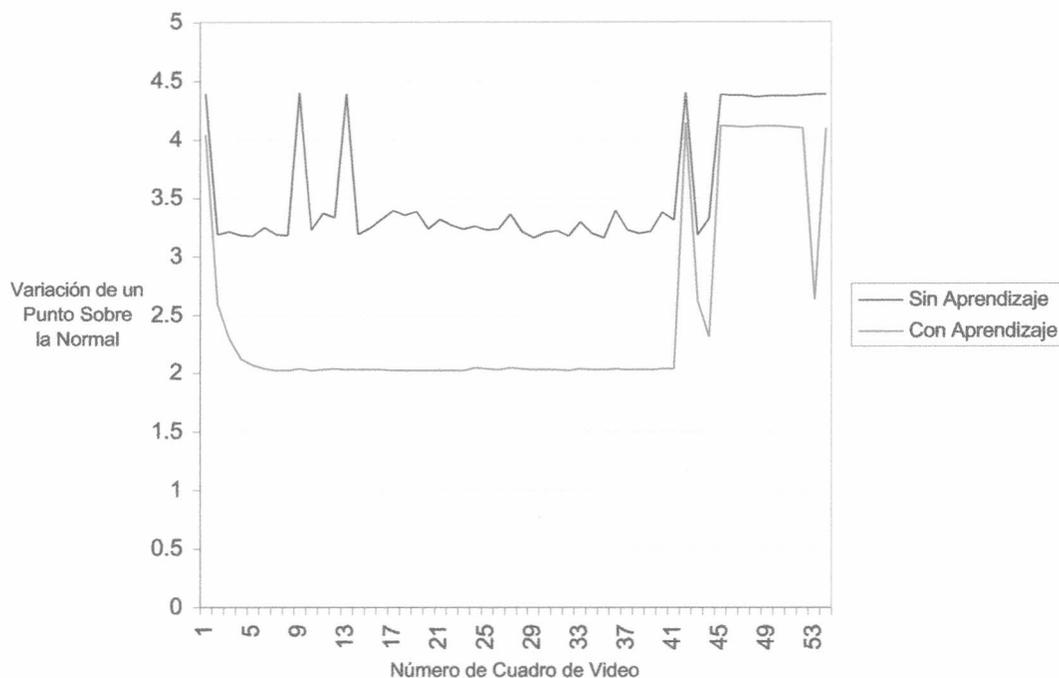


Figura 6-24. Gráfico que muestra los valores de $\rho_n^2(s)$ promedio alcanzados en cada cuadro, en una secuencia de video procesada con y sin modelo de estimación de movimiento.

En la Figura 6-24 se puede observar que los valores alcanzados por $\rho_n^2(s)$ son mayores cuando el movimiento del objeto no ha sido aprendido (no hay modelo de estimación de movimiento). Asimismo, se puede observar que la cantidad de veces que el sistema pierde al objeto (entra en *open loop*) es mayor cuando no se ha aprendido el movimiento (las entradas al *open loop* se pueden ver como los picos donde la curva alcanza su máximo). Esto se da porque el sistema no tiene certeza de donde debe ser ubicado el sector de interés a partir del cual se trazarán los segmentos normales. Por el contrario, se puede observar que cuando el modelo de seguimiento ha sido aprendido se mantiene estable y pierde al objeto muy pocas veces⁵⁶.

Un valor de $\rho_n^2(s)$ pequeño es útil en las aplicaciones de tiempo real, ya que implica que la longitud de los segmentos normales se reducirá (Recuérdese que la expresión para calcular la longitud de estos segmentos es *segmento normal* = $\bar{n}(s)\kappa\rho_n(s)$).

Asimismo, el valor de κ y el de la matriz de covarianza en el *open loop* pueden decrementarse (aunque esto se debe configurar manualmente).

⁵⁶ Excepto al final de la secuencia, donde como se explicó anteriormente, el objeto se deforma de manera no planar afín.

Una última cuestión que se puede observar de la Figura 6-24, es que cuando el sistema tiene un modelo de estimación de movimiento, el paso del *open loop* a un estado donde la incertidumbre es baja se da de manera exponencial (se puede ver ese salto en los primeros cuadros del video y entre el cuadro 42 y 45 de la Figura 6-24). En el caso en el cual el sistema no dispone de un modelo de estimación de movimiento, si bien existe un decrecimiento abrupto de la incertidumbre, nunca llega a ser tan notable. De esto se puede concluir que la recuperación del sistema es mucho mayor cuando el sistema cuenta con un modelo de estimación de movimiento.

En la siguiente sección se analizará en detalle los factores que afectan el desempeño del sistema de seguimiento.

6.5.4 Desempeño del Sistema

En esta sección se analizarán y compararán resultados obtenidos mediante la modificación de los parámetros de entrada del sistema de seguimiento. Estos resultados han sido obtenidos mediante el procesamiento de secuencias de video de 320 por 240 píxeles.

Para una mejor comprensión de los números y gráficos que aquí se muestran, se introducirán los siguientes términos:

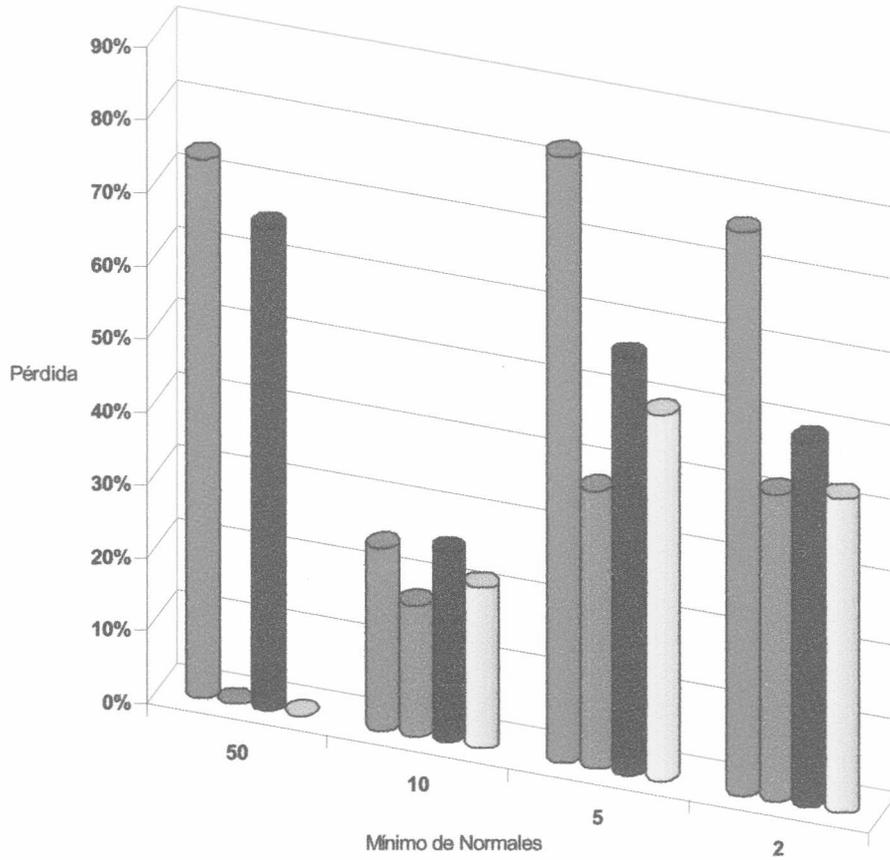
- Pérdida: Este parámetro mide la cantidad de cuadros de video perdidos⁵⁷ dividido por la cantidad de cuadros totales observados (expresado en porcentaje).
- Cantidad de cuadros por segundo (FPS): Indica los cuadros de video que son procesados por segundo.
- Eficiencia: La eficiencia es una métrica que combina los cuadros procesados por segundo sobre la cantidad porcentual de puntos perdidos. La fórmula empleada sería entonces:

$$FPS \left(\frac{\#Cuadros_Perdidos}{\#Cuadros_Observados} \right)$$

- Mínimo de normales: Es la longitud mínima que puede tener un segmento normal (cuando la certidumbre es máxima, los segmentos normales toman este valor.)
- SMEM: Sin Modelo de Estimación de Movimiento.
- CMEM: Con Modelo de Estimación de Movimiento.
- 9 pts., 17 pts., 34 pts.: Es la cantidad de puntos de control que posee la curva patrón empleada.

⁵⁷ Cada vez que se pierde un cuadro, el sistema entra al estado de *open loop*.

- Cantidad de cuadros omitidos: Es la cantidad de cuadros de video que el sistema en equilibrio dinámico omite al realizar el seguimiento.



	50	10	5	2
■ SMEM (17 pts)	74%	25%	83%	77%
■ CMEM (17 pts)	0%	18%	38%	42%
■ SMEM (34 pts)	66%	26%	57%	50%
□ CMEM (34 pts)	0%	22%	50%	43%

Figura 6-25. Mínimo de los normales vs Pérdida.

El gráfico de la Figura 6-25 muestra, a primera vista, que un sistema en el cual se han aprendido los parámetros de estimación de movimiento la pérdida es menor. Asimismo, se puede acortar más la longitud mínima de los segmentos normales, sin llegar a un sistema fácilmente desestabilizable. Tener normales de longitud mínima menor, lleva a un sistema que puede funcionar más rápidamente (como será visto en gráficos siguientes). En particular, cuando los segmentos normales tienen una longitud mínima de 50 el sistema se torna muy sensible al entorno, ya que ese tamaño de normales tiende a

capturar muestras que no pertenecen al objeto. Si el sistema no tiene valores de estimación de movimiento⁵⁸, le resulta muy difícil seguir al objeto correctamente.

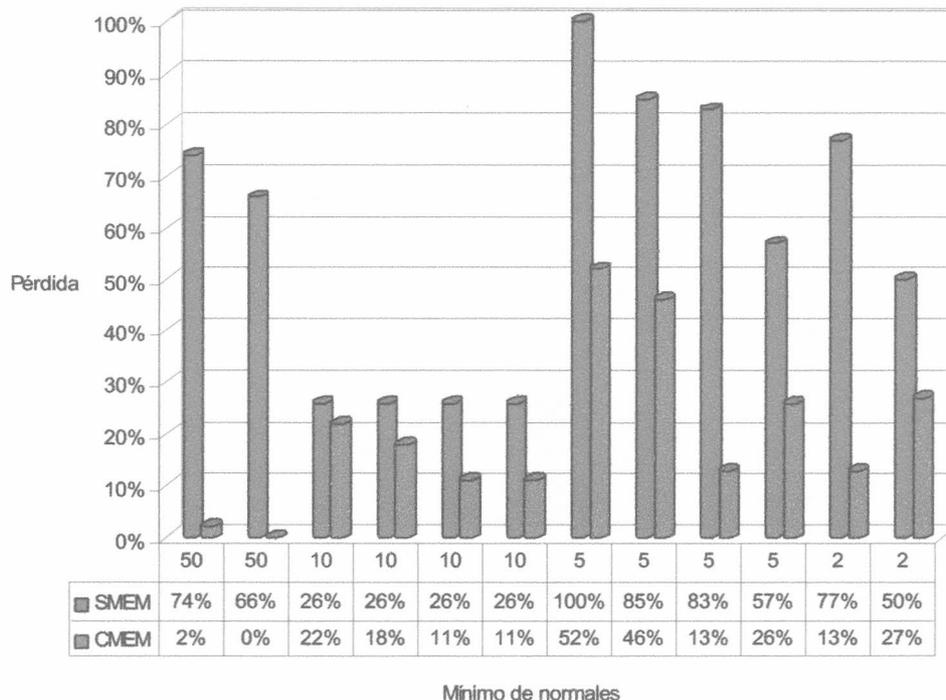


Figura 6-26. Sistema con modelo de estimación de movimiento y sin modelo de estimación de movimiento vs Pérdida.

La Figura 6-26, también muestra la pérdida en que incurre el sistema de seguimiento, dependiendo del modelo de estimación de movimiento y la longitud de los segmentos normales. Aquí se presenta un conjunto de experimentos realizados con el sistema de seguimiento en distintas configuraciones. Para hacer esto se tomó cada configuración y se analizó el caso con modelo de estimación de movimiento y sin modelo de estimación de movimiento. Aquí también se puede ver que cuando los segmentos normales son muy largos (50) y no se emplea el sistema de estimación de movimiento el sistema se torna muy inestable. Como se explicó anteriormente, este hecho se debe principalmente a que el sistema se confunde con el entorno. También se puede observar que existen configuraciones (sobre todo en la longitud igual a 10) donde el sistema se muestra bastante estable aún sin el modelo de estimación de movimiento. Esta configuración es la que debería ser empleada si se quiere usar el video para el aprendizaje de parámetros dinámicos. Asimismo, también se puede ver también que si se emplea el modelo de estimación de movimiento la pérdida es menor en cualquiera de las configuraciones presentadas.

⁵⁸ Recuérdese que se asume que el sistema no cuenta con valores de estimación de movimiento, si no se ha realizado un aprendizaje de parámetros dinámicos (a pesar de tener el modelo de velocidad constante cargado por defecto).

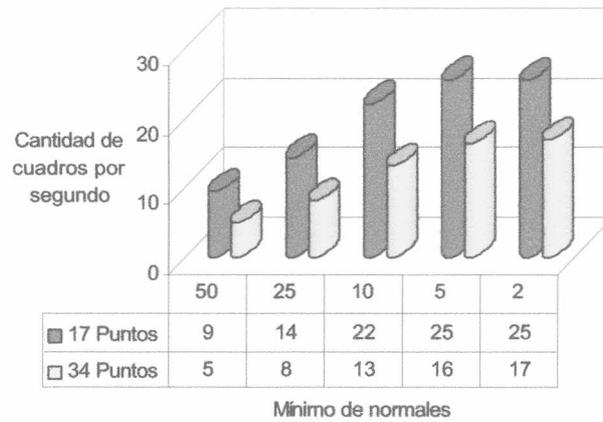


Figura 6-27. Mínimo de segmentos normales vs Cantidad de cuadros por segundo.

En la Figura 6-27 se puede observar lo que se ha comentado previamente: cuando la longitud mínima de los segmentos normales es menor, el sistema tarda menos en procesar cada cuadro. Cuando se establece una longitud mínima, esta debe ser recorrida incluso cuando la certidumbre del sistema es alta. Estableciendo la longitud mínima a valores pequeños, el sistema puede realizar ajustes más precisos al recorrido sobre el segmento normal, no recorriendo de más cuando no corresponde. Como contrapartida, cuando la longitud mínima es muy pequeña el sistema se desestabiliza con mayor facilidad, como puede verse en el gráfico de *mínimo de normales vs. pérdida* (en la sección 6.5.5 se mostrará por qué es necesario contar con un mínimo en la longitud de segmentos normales). Por último se puede observar como el incremento en la cantidad de puntos de control que conforman la curva patrón hace que se incremente el tiempo necesario para procesar cada cuadro. Si se quiere hacer un procesamiento de video en tiempo real, la cantidad de puntos de control de la curva patrón es un factor determinante.

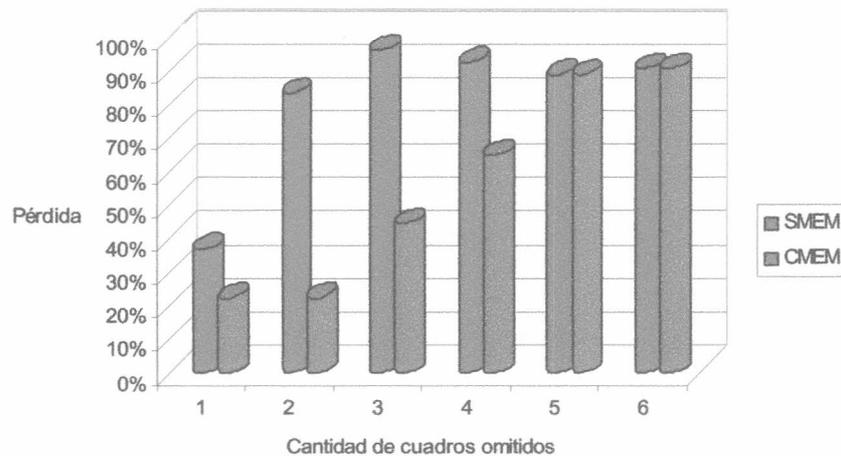


Figura 6-28. Cantidad de cuadros omitidos vs Pérdida

En la Figura 6-28 se puede ver la pérdida en que se incurre cuando se emplea el mecanismo de equilibrio dinámico. También se puede ver que en general este mecanismo debe ser empleado junto con un modelo de estimación de movimiento. También se puede

observar que la pérdida que sufre el sistema crece de manera abrupta llegado un cierto valor de cuadros omitidos. Esto se puede llegar a corregir realizando iteraciones de aprendizaje.

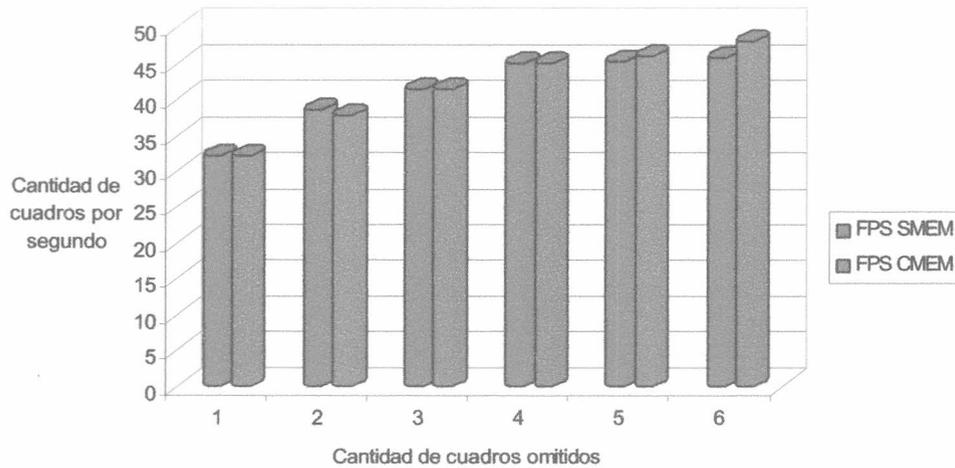


Figura 6-29. Cantidad de cuadros omitidos por el equilibrio dinámico vs Cantidad de cuadros por segundo

En el gráfico de la Figura 6-29 se puede observar la cantidad de cuadros por segundo que se pueden llegar a procesar dependiendo del número de cuadros omitidos por el equilibrio dinámico. Si bien los números son altos, este gráfico debe ser cotejado junto con el anterior, ya que la calidad del seguimiento baja considerablemente a medida que aumenta la cantidad de cuadros omitidos. Asimismo cuando la cantidad de cuadros llega a un cierto punto, no siempre se superará la cantidad de cuadros por segundo obtenidos con el valor inmediato anterior. Esto se encuentra ligado a la pérdida que sufre el sistema al emplear este mecanismo; si se pierden muchos puntos, entonces se estará saltando constantemente al *open loop*, lo cual degrada el desempeño del sistema de seguimiento (aquí es donde podría ser necesario un ajuste del valor en el cual se inicializan los segmentos normales en el *open loop*).

En la Figura 6-30 se observa como la cantidad de puntos de la curva patrón afecta la precisión con la que el sistema de seguimiento sigue al objeto. La cantidad de puntos en principio no determina que el seguimiento sea bueno o malo, sino cuan bien ajusta la curva resultado luego de la iteración del filtro. Por otro lado, una curva con menor cantidad de puntos de control es también una curva menos flexible, la cual tiene más problemas para adaptarse a un contorno que una curva con mayor cantidad de puntos. Esos problemas para adaptarse al contorno, podrían llevar en algunos casos a provocar una pérdida importante de puntos. En el caso que aquí se observa, las pequeñas diferencias se deben a este hecho. A pesar de todo, las diferencias se hacen más notables al observar los videos generados como resultado.

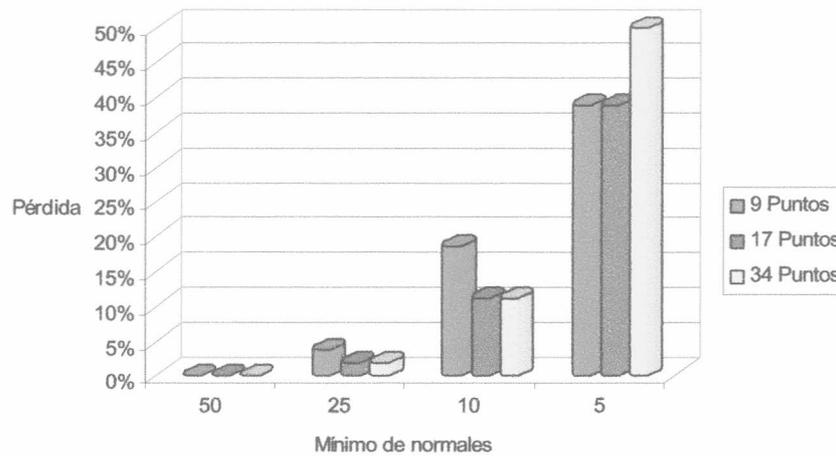


Figura 6-30. Mínimo de normales vs Pérdida, considerando distintas cantidades de puntos de control.

En la Figura 6-31 se puede ver que la eficiencia del sistema de seguimiento es mayor cuando la longitud de los segmentos normales alcanza el valor de 10. En este punto, el sistema encuentra un equilibrio entre cantidad de cuadros perdidos y velocidad de proceso. Este valor puede ser empleado para configurar el sistema de seguimiento para que trabaje con videos similares al empleado para el experimento, haciéndolo de la manera más eficiente posible.

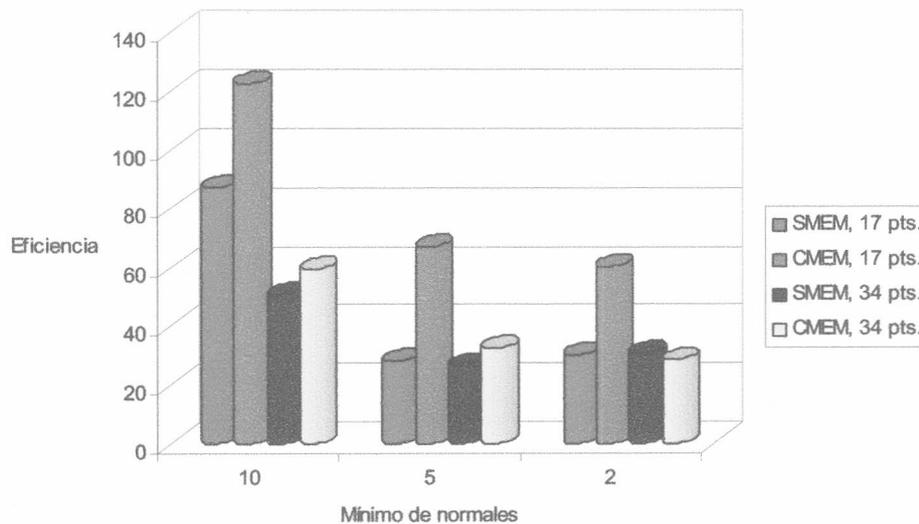


Figura 6-31. Mínimo de normales vs Eficiencia.

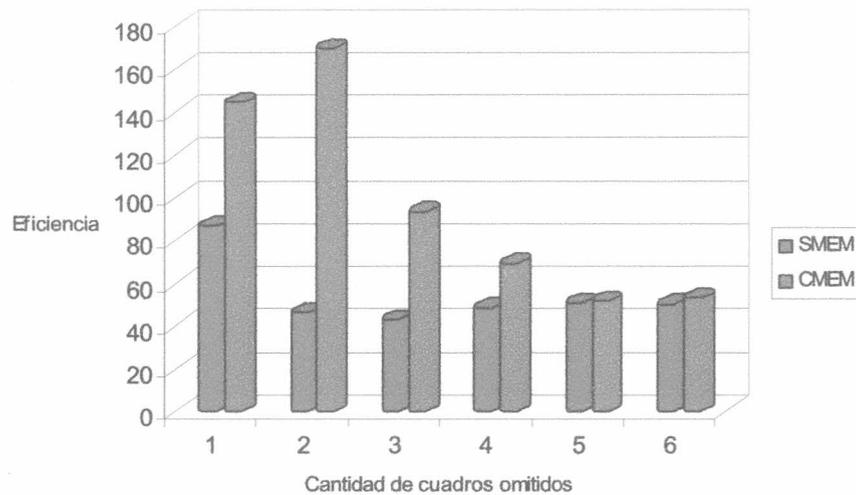


Figura 6-32. Cantidad de cuadros omitidos vs Eficiencia.

Finalmente, en la Figura 6-32 se puede ver que el sistema se comporta de manera más eficiente cuando la cantidad de cuadros omitidos es menor. La razón de esto, es que la cantidad de cuadros perdidos crece mucho más abruptamente que la cantidad de cuadros procesados por segundo, a partir de un cierto valor de cuadros omitidos.

6.5.5 Problemas encontrados

¿Por qué un mínimo en la longitud de los segmentos normales?

Al realizar el seguimiento de un objeto en una secuencia de video, se observa en cada cuadro la posición y forma del objeto y se hacen pequeñas correcciones respecto de la última posición conocida del mismo. Las observaciones propiamente dichas son el resultado del muestreo que se realiza a lo largo de los segmentos normales del sector de interés. Un problema que se ha encontrado al realizar el aprendizaje de los parámetros dinámicos del sistema de seguimiento⁵⁹ radica en la longitud que pueden llegar a tomar los segmentos normales. A medida que la certidumbre del sistema se incrementa, las longitudes de los segmentos normales decrecen alcanzando valores sumamente pequeños. Como consecuencia de este hecho, el sistema de seguimiento puede llegar a perder al objeto que está siguiendo y, debido a que no posee información previa del movimiento que realiza el objeto a seguir, el sistema no puede recuperarse de dicha pérdida.

Para solucionar el problema recientemente mencionado se utiliza un mínimo en la longitud de los segmentos normales. Este mínimo permite que el sistema de seguimiento pierda en menor medida al objeto que desea seguir.

⁵⁹ El hecho de estar realizando el aprendizaje de los parámetros dinámicos implica que el sistema no tiene información previa que permita ayudar a encontrar el objeto que se desea seguir.

El Problema del Anclaje

Para poder seguir más fácilmente a un objeto dentro de una secuencia de video puede ser útil anclar la curva patrón al objeto a seguir. Anclar una curva a un objeto significa que se identifican ciertos puntos especiales que permitirán recoger información relevante del movimiento realizado por el objeto.

Existen dos formas de resolver el problema del anclaje: la primera consiste en emplear algún elemento del objeto que se encuentra siguiendo el sistema (en la Figura 6-33, el dedo pulgar). Contorneando el dedo pulgar, este se convierte en un conductor importante de la transformaciones que sufre el contorno; si por ejemplo el objeto se encuentra desplazándose, entonces el sistema lo notará gracias al desplazamiento del dedo pulgar. Cuando no existe un elemento claro del objeto que pueda funcionar como conductor (no existe un elemento del objeto del cual *asir* el contorno), se puede trabajar con una mayor cantidad de puntos de control en algunos sectores. De esta manera, poner más puntos de control funciona como un *pegamento* entre la curva resultado y el objeto.

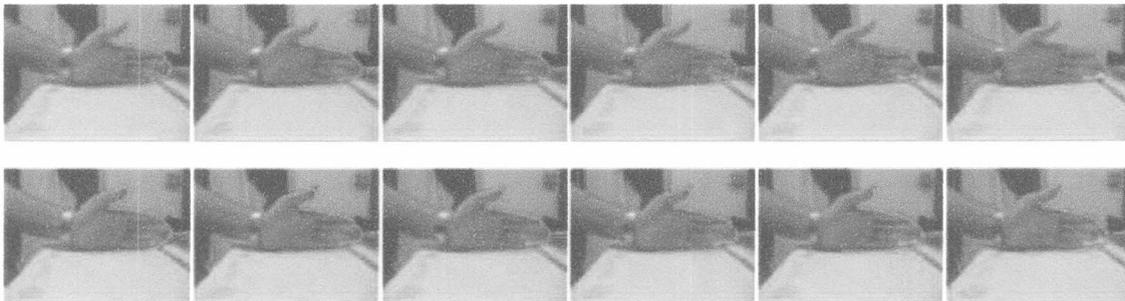


Figura 6-33. En la secuencia de video que se muestra en la parte superior, se puede ver un seguimiento realizado sin el anclaje necesario. La secuencia de video que se muestra en la parte inferior es equivalente, solo que aquí se ha contemplado una curva patrón con anclaje en el dedo pulgar

Es importante hacer una distinción de la forma en que se recorren las curvas a la hora de realizar el muestreo de los segmentos normales. Si la curva se recorre por longitud de arco (ver 6.2.1) puede suceder que no se esté prestando atención al anclaje deseado⁶⁰ y por consiguiente se pierda al objeto que se deseaba seguir. Si en cambio, la curva se recorre incrementando en una cantidad fija el valor del parámetro, se obtendrá una mayor cantidad de muestras donde se encuentre una mayor concentración de puntos de control. Este último es el enfoque que es necesario emplear cuando se quieren usar anclajes.

Manejando las Sombras

La sombra es un *objeto* del entorno, que sigue muy de cerca al objeto sobre el que trabaja el sistema de seguimiento. Por lo general, se manifiesta como un oscurecimiento del fondo de la escena y, en la mayoría de los casos se logra eliminar mediante ajustes a la tolerancia de la substracción del fondo. Existen casos donde ajustar la tolerancia puede no ser suficiente, pero se pueden tratar haciendo ciertas consideraciones:

⁶⁰ Si por ejemplo el anclaje ha sido realizado mediante zonas donde existe una mayor concentración de puntos de control, el muestreo por longitud de arco anula totalmente dicho efecto.

1) Si se trabaja en exteriores, puede ser necesaria la consideración de un horario específico de forma de que la sombra se encuentre lejos del objeto seguido por el sistema; trabajar al mediodía donde la sombra se encuentra muy cerca del objeto seguido por el sistema, no debería presentar inconveniente, ya que la sombra es casi igual al objeto. Asimismo, emplear videos filmados al atardecer hará que la sombra se estire y se aleje del objeto (sobre todo en otoño e invierno). Si lo que se sigue es el torso de la persona, en principio no debería haber inconveniente. Por otro lado, si se considera un video filmado en un horario en el cual la sombra se encuentra lejos del objeto, pero lo suficientemente cerca como para que los segmentos normales de búsqueda la encuentren, entonces el sistema realizará un seguimiento erróneo.

2) Si la fuente de luz es puntual⁶¹ y el fondo de la escena es de color claro, la sombra se encontrará muy definida y no desaparecerá ajustando la tolerancia. En casos como este hay que considerar la colocación de otra fuente de luz de manera de atenuar el color que la sombra proyecta sobre el fondo de la escena. Asimismo, las consideraciones respecto a la posición de la fuente de luz realizadas en el caso anterior también pueden ser aplicadas al caso de luz artificial.

Sistema reactivo vs. Aprendizaje

Se ha visto en los ejemplos mostrados en la sección 6.5.2, que el sistema se mantiene estable aún en presencia de oclusiones cuando los parámetros de la estimación de movimiento han sido configurados. Esto se puede traducir en que el sistema le asigna un mayor peso a la estimación que a la observación. Una consecuencia de esto, es que el sistema perderá reacción. Si el objeto que se encuentra siguiendo no respeta el modelo de estimación de movimiento, al sistema le será más complicado estabilizarse y el tiempo que tome en hacerlo puede hacer que se pierda al objeto seguido. Esto ocurrirá porque la estimación indicará un movimiento hacia un sector (indicado por el modelo de estimación de movimiento), mientras que el objeto se dirige hacia otro. Tan pronto como se halla evidenciado la pérdida del objeto, el sistema disparará los mecanismos habituales para el caso (estiramiento de los segmentos normales, etc), aunque puede ser que para entonces el objeto se haya ido del alcance. Asimismo, si el sistema logra recuperarse de la desestabilización, se encontrará funcionando con un modelo de estimación de movimiento que le dará más complicaciones que ayuda, ya que las estimaciones serán erróneas constantemente.

6.6 Mejoras posibles y trabajo a futuro

6.6.1 Otros filtros para muestreo con segmentos normales

Cuando no se dispone de un fondo de escena fijo, así como tampoco del movimiento realizado por la cámara (el cual puede ser aleatorio), lo mejor es emplear algún tipo de filtro como Canny, Roberts, Sobel, etc [28]. Este tipo de filtros de detección de bordes no tienen el problema de necesitar una cámara predeterminada, aunque en situaciones donde el entorno contiene mucho ruido los resultados pueden no llegar a ser buenos.

⁶¹ Por ejemplo una bombita de luz.

Asimismo, el costo computacional de la aplicación de estos filtros sobre cada normal excede al costo que tiene la aplicación de la substracción de fondo en la forma en la que se aplica en la presente implementación⁶². En cualquier caso, el sistema siempre necesitará un modelo de estimación de movimiento para mejorar el seguimiento.

6.6.2 Muestreo con segmentos normales: eliminando el cuello de botella

En la presente implementación el muestreo de segmentos normales se realiza con el algoritmo DDA. Este algoritmo muestrea los píxeles uno por uno, por toda la longitud del segmento normal hasta encontrar una muestra. Esto hace que el módulo del sistema encargado del sampleo de segmentos normales del sistema, sea un cuello de botella. Es posible sin embargo mejorar la implementación presentada de forma tal que se realice un muestreo como en [4]. Allí se muestrea la normal en intervalos espaciados, con el consiguiente ahorro de cálculos. Sin embargo con este sistema se pierde precisión, ya que si un punto de la normal cae muy cerca del borde del objeto, en el próximo paso la normal puede llegar a penetrar muy adentro del objeto. De esta forma el punto de la curva característica queda ubicada dentro del objeto, por lo que puede llevar a una aproximación errónea de esta curva.

Un pseudo código para realizar el muestreo de los segmentos normales de a pasos se puede ver en la Figura 6-34.

```
function samplearNormal, p1, p2
  pInicio = p1
  direccion = p2 - p1
  cantMuestras = CantidadDeMuestrasATomarSobreElSegmentoNormal
  for i = 0, cantMuestras do begin
    p1 = pInicio + ( i / cantMuestras ) * direccion
    if( encuentraMuestra( p1 ) ) then begin
      return, p1
    endif
  endfor
  return, idMuestraNoEncontrada
end; samplearNormal
```

Figura 6-34. Pseudo código para un algoritmo de muestreo de segmentos normales de a pasos.

⁶² En una aplicación de tiempo real como la presente, cualquier operación que se pueda ahorrar es importante para mejorar el desempeño del sistema.

El algoritmo cuyo pseudo-código es el de la Figura 6-34, toma como parámetro dos puntos y determina si puede encontrar alguna muestra⁶³ en el segmento que determinan. Para hacer esto, se discretiza la recta en una cierta cantidad de puntos determinada por una constante del sistema y se verifica si se ha encontrado una muestra en cada punto que conforma la discretización.

6.6.3 Mejorando el modelo: El trabajo con gran cantidad de curvas

Una mejora inmediata que surge a partir de la sección 6.4-Trabajo con Varias Curvas, consiste en segmentar aún más de las curvas principales, de forma de lograr un conjunto mayor de curvas más pequeñas. El sentido de ésta modificación, sería la representación de un modelo del cuerpo humano que pueda manejar deformaciones mucho más complejas que las provistas por el espacio planar afín. Un modelo como éste, podría ser de gran ayuda en las aplicaciones que requieren de captura de movimiento [31], animación de personajes, etc. Sin embargo, debe tenerse en cuenta que las complicaciones aumentan a medida que aumenta el número de curvas, ya que al sistema de seguimiento le cuesta cada vez más discriminar qué parte del objeto corresponde a cada curva, llegando a ser crítico el problema si se presentan auto-oclusiones.

De cualquier manera, sea cual fuere la solución que se elija adoptar para manejar una gran cantidad de curvas operando sobre el mismo objeto, es seguro que el sistema deberá apoyarse fuertemente en un modelo de estimación de movimiento.

6.6.4 Multiprocesamiento y Procesamiento en Paralelo

Si bien la situación ideal es lograr un sistema de seguimiento que funcione sin necesidad de un hardware costoso, siempre resulta de interés una implementación que trabaje en paralelo, sobre todo considerando que existe una cantidad importante de operaciones secuenciales, que podrían realizarse en paralelo. Por ejemplo, los muestreos de normales en una iteración k del filtro de Kalman se pueden realizar de manera independiente, ya que cada punto muestreado no tiene relación con el punto siguiente o el anterior.

En [13] por ejemplo, se puede observar una implementación para multiprocesador, la cual considera el procesamiento de manera independiente en cada *span* de la spline.

6.6.5 Múltiples curvas patrón

Una curva patrón no se puede deformar adecuadamente cuando el objeto que contornea realiza una transformación no contemplada en el espacio de formas del sistema de seguimiento. Sin embargo, se puede hacer una adaptación para dar más libertad al

⁶³ Se considera muestra a todo punto de la imagen que difiere de la imagen de fondo, según la tolerancia utilizada (referirse a la sección 3.1.2 donde se comenta el mecanismo de substracción de fondo).

espacio de formas, sin necesidad de considerar una mayor cantidad de grados de libertad⁶⁴.

Supóngase que se considera el espacio de formas planar afín. La idea es que cuando se detecta una transformación no contemplada en dicho espacio, exista una curva patrón que pueda seguir al objeto haciendo uso sólo de transformaciones planares afines. De esta manera existirían tantas curvas patrón como transformaciones no contempladas en el espacio de formas realice el objeto. Esto se puede ver en la Figura 6-35.

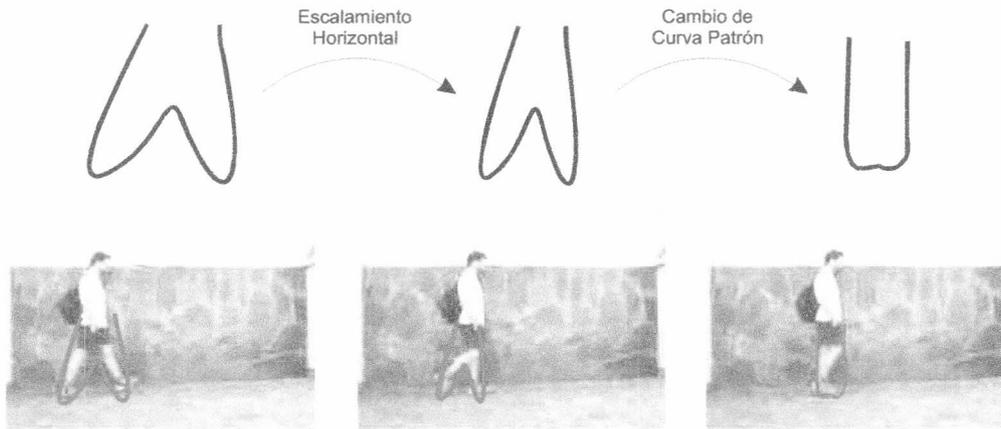


Figura 6-35. Esquema de la aplicación de múltiples curvas patrón a una secuencia de video de una persona caminando.

El problema de detectar cuando un objeto se encuentra realizando una transformación no planar afín no es simple y dependerá del caso. En el caso de una persona caminando se puede tomar la distancia entre algún par de puntos de la curva patrón, de manera que cuando se encuentren a una determinada distancia se realice el cambio. Esto se puede ver en la Figura 6-36.

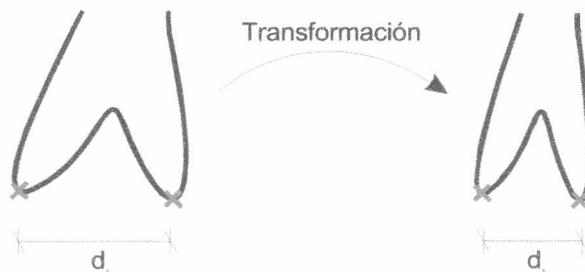


Figura 6-36. Esquema de detección de cambio de curva patrón. Cuando los puntos indicados con una cruz se encuentran a una distancia menor que d_1 , se dispara en cambio de curva patrón.

⁶⁴ Recuérdese que las dimensiones de las matrices intervinientes en los cálculos realizados por el sistema, se encuentran en función de las dimensiones del espacio de formas elegido. Agrandar las matrices hará que el sistema tenga que realizar un proceso de cálculo mucho mayor.

Finalmente, las curvas patrón deben funcionar en paralelo de manera de que cualquiera de ellas esté siempre lista para reemplazar a la curva en uso; esto implica que de alguna forma las transformaciones aplicadas a la curva en uso, se deben aplicar también a las otras curvas patrón. Si se considera el caso de las traslaciones, el problema se reduce a trasladar todas las curvas juntas; si otro tipo de deformaciones es contemplado entonces el problema es complicado y dependerá de cada caso particular.

6.6.6 Encontrando el sector inicial

Sería interesante que el sector inicial a partir del cual se realice el seguimiento se pudiera armar automáticamente, para poder prescindir de la interacción del usuario como en la presente implementación⁶⁵. Existen métodos que permiten buscar donde se encuentra un objeto en un cuadro de una secuencia de video. Muchos de estos métodos se basan en regiones y consisten en analizar una región de un cuadro de video e identificar alguna particularidad como por ejemplo una distribución particular de píxeles. En el presente trabajo, donde se emplea la substracción del fondo, una *distribución particular de píxeles* se puede traducir en *píxeles de un color distinto de negro*. Esa región de píxeles coloreados se puede identificar como el objeto que se desea seguir. De más está decir que se deben tomar recaudos para no confundir al sistema con algún otro objeto que se encuentre moviendo en la escena o con algún residuo de la substracción del fondo. A partir de la información obtenida, se debería analizar que traslación aplicarle a la curva patrón para ubicarla en ese sector y comenzar con el seguimiento. Para entender un poco más el funcionamiento de mecanismos de este tipo, referirse [15] donde se puede ver un modelo basado en variables fotométricas.

6.6.7 Substracción del fondo sin cámara fija

Una de las restricciones más importantes que plantea el sistema de seguimiento implementado es la estaticidad de la cámara. Esta estaticidad es necesaria debido a la substracción del fondo que se realiza en cada cuadro de la imagen. Si bien se podría encontrar el objeto a seguir mediante filtros convencionales para detección de bordes (Canny, Roberts, Sobel, etc.) [28], la substracción del fondo presenta una cantidad de ventajas que se consideran importantes (las cuales fueron vistas en 3.1.1). El esquema a considerar, sería uno en el cual se realice substracción del fondo, a la vez que la cámara realiza movimientos. A primera vista parece que tal esquema es contradictorio, debido a que el método clásico de substracción de fondo toma una imagen una única vez y compara todos los cuadros con ésta. Sin embargo, si el movimiento de la cámara se encuentra bien definido (por ejemplo, un giro de X cantidad de grados en el plano perpendicular a la imagen) se puede llegar a predecir con bastante precisión cuál será el nuevo fondo en cada momento. Este estará compuesto por el fondo del cuadro anterior, mas un fragmento de la nueva imagen correspondiente a la X cantidad de grados girados por la cámara. Este mecanismo tiene ciertas restricciones que se deben cumplir, como por ejemplo, al momento de incorporar un nuevo fragmento del fondo es necesario que no exista un objeto que luego se moverá de su ubicación. Sin embargo en aplicaciones

⁶⁵ Recuérdese que en la presente implementación, se le indica al sistema donde se debe ubicar el primer sector de interés a partir del cual se comienza a realizar el seguimiento.

donde el propósito de la cámara sea seguir a un individuo (una cámara de seguridad, un partido de fútbol, etc.) este método podría ser útil.

6.6.8 Un sistema inteligente de seguimiento

Uno de los objetivos que muchas veces se persigue en las aplicaciones científicas es el de lograr sistemas que cada vez requieran menos intervención humana. En el contexto de un sistema de seguimiento implementado como en el presente trabajo, requerir menos intervención humana significa que el sistema no solo ubique al objeto a seguir, sino que pueda establecer sus parámetros de acuerdo al historial del seguimiento que se encuentra realizando. La idea es que el sistema monitoree constantemente el rendimiento que va teniendo junto con la certidumbre. Sobre la base de estos indicadores el sistema debería ser capaz de ajustar los parámetros dinámicos del modelo⁶⁶, la longitud (mínima y en *open loop*) de los segmentos normales y también la cantidad de cuadros a omitir en el estado de equilibrio dinámico.

Idealmente, lo que se desearía es poseer un sistema al que se le ingrese la curva patrón a buscar, éste encontrara en la secuencia de video cual es el objeto que mejor se ajusta a la curva parámetro y a partir de ese momento realizar el seguimiento.

6.6.9 Análisis de Componentes Principales (PCA)

Se puede emplear este método, con el objeto de reducir la magnitud de algunos elementos involucrados en los cálculos.

La matriz de covarianza \bar{P} tiene una dimensión de $N_x \times N_x$ ⁶⁷. La dimensión de esta matriz depende de la dimensión del espacio de formas utilizado, el cual a su vez depende de la cantidad de grados de libertad del espacio de formas. Si la cantidad de tipos⁶⁸ de movimiento realizados por el objeto a seguir es muy inferior a la cantidad de tipos de movimiento de que dispone el espacio de formas considerado, no solo \bar{P} se acercará a la singularidad, sino que además se estará operando con matrices que son innecesariamente grandes⁶⁹.

Si en vez de considerar un espacio de formas genérico (similitudes Euclidianas, planar afín, etc.) se considera un espacio de formas que pueda generar transformaciones para representar sólo los tipos de movimiento realizados por el objeto que se está siguiendo,

⁶⁶ Estos parámetros se podrían estimar por extrapolación de acuerdo a las observaciones realizadas en tiempo de aprendizaje.

⁶⁷ Donde X es la cantidad de grados de libertad del espacio de formas considerado.

⁶⁸ Según se ha mencionado anteriormente, en el contexto del presente trabajo se cuenta con seis tipos de movimiento (de acuerdo al espacio planar afín): rotación, traslación vertical, traslación horizontal, escalamiento horizontal, escalamiento vertical y escalamiento diagonal.

⁶⁹ Así como la matriz de covarianza, hay una gran cantidad de elementos que emplea el algoritmo de ajuste, que deben su dimensión al tamaño del espacio de formas considerado.

se puede mejorar⁷⁰ el rendimiento del algoritmo de ajuste utilizando el análisis de componentes principales.

Básicamente, el algoritmo de PCA construye una matriz W y una curva Q_0' (cuyas dimensiones se espera que sean menores a $N_x \times N_x$ y $N_Q \times 1$ respectivamente), de acuerdo a un conjunto de muestras de prueba (las muestras de prueba son tomadas con los grados de libertad existentes algún espacio de formas genérico). Para un mayor detalle acerca del método ver [5].

En el presente trabajo, las transformaciones empleadas para seguir al cuerpo humano emplean, por lo general, todo el espectro de posibilidades de que dispone el espacio planar afín. Es por esto que no se ha hecho hincapié en el uso de un método que restrinja los grados de libertad, tal como el PCA.

6.7 Conclusiones finales

Se ha visto al sistema en funcionamiento y los resultados han sido satisfactorios. El sistema se ha mostrado robusto en la mayoría de los casos probados, manejando incluso situaciones en las que las transformaciones que sufría el objeto caían fuera del espacio de formas planar afín. Su desempeño en tiempo real ha sido aceptable, llegando a valores más allá de los 24 cuadros por segundo, merced al mecanismo de equilibrio dinámico funcionando en conjunto con el modelo de estimación de movimiento. Este mismo modelo de estimación de movimiento mostró ser una pieza fundamental cuando el objeto seguido se vio frente a una oclusión. El modelo de estimación de movimiento es una ayuda fundamental con la que debe contar un sistema de seguimiento.

Son particularmente interesantes las mejoras que se podrían hacer al sistema para mejorar su rendimiento en *cuadros por segundo*. Al tener a disposición una ventana mayor de tiempo para procesar cada cuadro es posible hacer ajustes para mejorar el resultado obtenido en cada iteración⁷¹. Entre las mejoras que podrían hacerse con el objeto de mejorar la velocidad de procesamiento, el manejo que los segmentos normales es crítico, ya que como se ha mencionado anteriormente, se encuentra allí uno de los cuellos de botella más importantes que presenta el sistema.

Elevando el tiempo disponible para procesar cada cuadro se pueden hacer ajustes interesantes al resultado agregando módulos de cálculo de pose y perspectiva⁷² (para más detalle ver [5]). En [3] se puede ver un sistema que luego de realizar la iteración correspondiente del filtro de Kalman, realiza un conjunto de iteraciones en un proceso denominado *refinement*, con el objeto de ajustar aún más el resultado.

⁷⁰ Esta mejora en principio dependerá del espacio de formas que se consideró inicialmente y del tipo de movimientos que realiza el objeto. Por ejemplo si se considera el espacio de formas planar afín y el objeto realiza sólo traslaciones, la ganancia es un recorte de 4 dimensiones.

⁷¹ Si el sistema no posee tiempo para realizar cálculos extras, no puede refinar el resultado obtenido (considerando que se quiere mantener funcionando el sistema en tiempo real).

⁷² Durante el transcurso del trabajo se evidenció algún pequeño defecto de seguimiento que podría haber sido corregido por un módulo de cálculo de perspectiva.

Entre las mejoras que pueden hacerse para flexibilizar el modelo, sin el agregado de complejidad de cálculo adicional, resulta destacable la implementación que maneja un conjunto de curvas que siguen al mismo objeto. El ejemplo mostrado con el sistema que se ha implementado muestra un poco del potencial que presenta la mejora. Si se puede representar cada miembro del cuerpo humano con una curva diferente y se pueden manejar temas como la auto-oclusión, entonces la representación de una persona con curvas se podrá realizar con un sistema simple y poco costoso computacionalmente.

Desde el punto de vista de la utilización, se podría decir que aún queda mucho por implementar si lo que se quiere es una aplicación versátil y que pueda ser empleada por el usuario común, en una amplia gama de circunstancias. El trabajo con las curvas patrón es tedioso y requiere de una comprensión importante del problema y de las restricciones que presenta el sistema de seguimiento. La forma, cantidad y distribución de puntos que tiene una curva patrón hace la diferencia entre un sistema de seguimiento que funciona de manera robusta y eficiente y un sistema de seguimiento que no es capaz de seguir al objeto que desea. Técnicas como el anclaje, distribución de puntos de control, orientación de la curva deben ser bien comprendidas para lograr que una curva patrón haga que el sistema se comporte de manera exitosa.

El sistema cuenta además con una importante cantidad de parámetros que pueden ser combinados de muchas maneras para adaptar al sistema al entorno en el que debe trabajar. Esa configuración requerirá, casi con seguridad, experimentar con un conjunto de secuencias de aprendizaje, las cuales deben ser bien elegidas o el modelo de estimación de movimiento no corresponderá con el movimiento que realiza el objeto que se desea seguir. En definitiva, existe una gran cantidad de parámetros, consideraciones y restricciones que se deben tener en cuenta a la hora de hacer funcionar el sistema y solo comprendiendo el problema en su totalidad podrá un usuario configurar al sistema de seguimiento para que trabaje de manera eficiente en el entorno que desee.



Capítulo 7: Apéndice A

7.1 ¿Por qué funciona el algoritmo de minimización propuesto?

El objetivo de esta sección es mostrar por qué el algoritmo propuesto en la Figura 3-6 devuelve la transformación X tal que minimiza la ecuación (3-3). Para realizar la demostración se considera lo siguiente

$$S_0 = 0, Z_0 = 0 \text{ y se cumple que}$$

$$\begin{cases} v_i &= (r_f(s_i) - r(s_i)) \cdot \bar{n}(s_i) \\ h(s_i)^T &= \bar{n}(s_i)^T U(s_i) W \\ S_i &= S_{i-1} + \frac{1}{\sigma_i^2} h(s_i) h(s_i)^T \\ Z_i &= Z_{i-1} + \frac{1}{\sigma_i^2} h(s_i) v_i \end{cases}$$

Se puede ver que S , Z y h son las construidas por el algoritmo de la Figura 3-6. Partiendo de esta consideración se probará por inducción en i que

$$\text{Dado } T_i = \sum_{j=1}^i \frac{1}{\sigma_j^2} (v_j - h(s_j))^T [X - \bar{X}]^2, \text{ entonces}$$

$$T_N = (X - \bar{X})^T S_N (X - \bar{X}) - (X - \bar{X})^T Z_N - Z_N^T (X - \bar{X}) + C_N$$

Sea $i=1$

$$T_1 = \sum_{j=1}^1 \frac{1}{\sigma_j^2} (v_j - h(s_j))^T [X - \bar{X}]^2 = \frac{1}{\sigma_1^2} (v_1 - h(s_1))^T [X - \bar{X}]^2$$

$$T_1 = \frac{1}{\sigma_1^2} (v_1 - h(s_1))^T [X - \bar{X}]^T (v_1 - h(s_1))^T [X - \bar{X}]$$

$$T_1 = \frac{1}{\sigma_1^2} [v_1^T v_1 - [X - \bar{X}]^T h(s_1) v_1 - v_1^T h(s_1)^T [X - \bar{X}] + [X - \bar{X}]^T h(s_1) h(s_1)^T [X - \bar{X}]]$$

$$T_1 = \frac{1}{\sigma_1^2} v_1^T v_1 - [X - \bar{X}]^T \underbrace{\frac{1}{\sigma_1^2} h(s_1) v_1}_{Z_1} - \underbrace{\frac{1}{\sigma_1^2} v_1^T h(s_1)^T}_{Z_1^T} [X - \bar{X}] + [X - \bar{X}]^T \underbrace{\frac{1}{\sigma_1^2} h(s_1) h(s_1)^T}_{S_1} [X - \bar{X}]$$

La expresión resultante hace que el caso base quede probado

$$T_1 = \underbrace{\frac{1}{\sigma_1^2} v_1^T v_1}_{\text{Término Constante}} - (X - \bar{X})^T Z_1 - Z_1^T (X - \bar{X}) + (X - \bar{X})^T S_1 (X - \bar{X})$$

Siguiendo con la inducción se asume ahora que vale $T_{N-1} = (X - \bar{X})^T S_{N-1} (X - \bar{X}) - (X - \bar{X})^T Z_{N-1} - Z_{N-1}^T (X - \bar{X}) + C_{N-1}$ y se desea ver si vale para T_N .

$$\begin{aligned} T_N &= \sum_{j=1}^N \frac{1}{\sigma_j^2} (v_j - h(s_j))^T [X - \bar{X}]^2 = \\ &= \sum_{j=1}^{N-1} \frac{1}{\sigma_j^2} (v_j - h(s_j))^T [X - \bar{X}]^2 + \frac{1}{\sigma_N^2} (v_N - h(s_N))^T [X - \bar{X}]^2 \stackrel{H.I.}{=} \\ &\stackrel{H.I.}{=} (X - \bar{X})^T S_{N-1} (X - \bar{X}) - (X - \bar{X})^T Z_{N-1} - Z_{N-1}^T (X - \bar{X}) + C_{N-1} + \\ &\quad + \frac{1}{\sigma_N^2} (v_N - h(s_N))^T [X - \bar{X}]^2 = \\ &= (X - \bar{X})^T S_{N-1} (X - \bar{X}) - (X - \bar{X})^T Z_{N-1} - Z_{N-1}^T (X - \bar{X}) + C_{N-1} + \\ &\quad + \frac{1}{\sigma_N^2} (v_N - h(s_N))^T [X - \bar{X}]^2 (v_N - h(s_N))^T [X - \bar{X}] = \\ &= (X - \bar{X})^T S_{N-1} (X - \bar{X}) - (X - \bar{X})^T Z_{N-1} - Z_{N-1}^T (X - \bar{X}) + C_{N-1} + \\ &\quad + \frac{1}{\sigma_N^2} v_N^T v_N - [X - \bar{X}]^T \underbrace{\frac{1}{\sigma_N^2} h(s_N) v_N}_{Z_N - Z_{N-1}} - \underbrace{\frac{1}{\sigma_N^2} v_N^T h(s_N)^T}_{Z_N^T - Z_{N-1}^T} [X - \bar{X}] + \\ &\quad + [X - \bar{X}]^T \underbrace{\frac{1}{\sigma_N^2} h(s_N) h(s_N)^T}_{S_N - S_{N-1}} [X - \bar{X}] = \\ &= (X - \bar{X})^T S_{N-1} (X - \bar{X}) - (X - \bar{X})^T Z_{N-1} - Z_{N-1}^T (X - \bar{X}) + C_{N-1} + \\ &\quad + \frac{1}{\sigma_N^2} v_N^T v_N - (X - \bar{X})^T (Z_N - Z_{N-1}) - (Z_N^T - Z_{N-1}^T) (X - \bar{X}) + \\ &\quad + (X - \bar{X})^T (S_N - S_{N-1}) (X - \bar{X}) = \\ &= (X - \bar{X})^T S_{N-1} (X - \bar{X}) - (X - \bar{X})^T Z_{N-1} - Z_{N-1}^T (X - \bar{X}) + C_{N-1} + \\ &\quad + \frac{1}{\sigma_N^2} v_N^T v_N - (X - \bar{X})^T Z_N + (X - \bar{X})^T Z_{N-1} - Z_N^T (X - \bar{X}) + Z_{N-1}^T (X - \bar{X}) + \\ &\quad + (X - \bar{X})^T S_N (X - \bar{X}) - (X - \bar{X})^T S_{N-1} (X - \bar{X}) = \\ &= C_{N-1} + \underbrace{\frac{1}{\sigma_N^2} v_N^T v_N}_{\text{Término constante } C_N} - (X - \bar{X})^T Z_N - Z_N^T (X - \bar{X}) + (X - \bar{X})^T S_N (X - \bar{X}) \end{aligned}$$

Con lo cual queda demostrado que

$$T_N = (X - \bar{X})^T S_N (X - \bar{X}) - (X - \bar{X})^T Z_N - Z_N^T (X - \bar{X}) + C_N$$

Sea ahora $T = (X - \bar{X})^T S (X - \bar{X}) + T_N$. Dado que se busca el X que minimiza T , lo que se hace es derivar e igualar a 0. Usando lo que se acaba de probar, queda entonces que

$$\nabla T = \nabla \left((X - \bar{X})^T S (X - \bar{X}) + (X - \bar{X})^T S_N (X - \bar{X}) - (X - \bar{X})^T Z_N - Z_N^T (X - \bar{X}) + C_N \right) = 0 \quad (7-1)$$

En este punto se emplearán las siguientes propiedades

- a) $(X - \bar{X})^T Z = Z^T (X - \bar{X})$, ya que en ambos lados se realiza un producto escalar.
- b) $\left[(X - \bar{X})^T (S + S_N) \right]^T = (S + S_N) (X - \bar{X})$, ya que tanto S_N como S son matrices simétricas⁷³.
- c) $\nabla (X^T A X) = 2X^T A$

De esta manera, rescribiendo (7-1) y empleando las propiedades antes enunciadas se obtiene que

$$\begin{aligned} \nabla T &= \underbrace{2(X - \bar{X})^T S}_{\text{Usando c)}} + \underbrace{2(X - \bar{X})^T S_N}_{\text{Usando c)}} - \underbrace{2Z_N^T}_{\text{Usando a)}} = 0 \\ 2(X - \bar{X})^T (S + S_N) - 2Z_N^T &= 0 \quad \{ \text{Transponiendo a ambos lados} \\ \left(2(X - \bar{X})^T (S + S_N) \right)^T &= (2Z_N^T)^T \quad \{ \text{Usando b)} \\ 2(S + S_N) (X - \bar{X}) &= 2Z_N \end{aligned}$$

Finalmente se despeja X quedando

$$X = \bar{X} + (S + S_N)^{-1} Z_N \quad (7-2)$$

Dado que se ha demostrado que T es equivalente a la ecuación (3-13), el X obtenido como resultado de la minimización de una también minimiza la otra.

De acuerdo a las consideraciones enunciadas previamente, S y Z son construidas en el paso n -ésimo del algoritmo de la Figura 3-6; \bar{X} y S son datos de entrada del mismo. Dado que (7-2) es lo que resuelve el algoritmo de ajuste en su paso 6), se puede ver que el valor de X que devuelve el mismo es el que minimiza la ecuación (3-3).

⁷³ Ambas matrices son inversas de matrices de covarianza; dado que las matrices de covarianza son simétricas, su inversa también será simétrica.

7.2 Demostración del crecimiento ilimitado de la varianza

Supóngase que se tiene una partícula, la cual se va desplazando constantemente una cantidad fija y en la misma dirección. Se puede considerar la siguiente desigualdad:

$$\sum_{i=0}^{N-1} (x_i - \bar{x})^2 < (N-1)(x_N - x)^2 \quad (7-3)$$

Supóngase que cada x_i representa una posición de la partícula. Dado el desplazamiento constante y siempre en la misma dirección, se puede asumir que $|x_i| < |x_{i+1}|$ (con lo cual se cumple el caso particular de (7-3), donde $\forall i: |x_i| < |x_n|$).

Si se continúa operando con la expresión se obtiene

$$\frac{\sum_{i=0}^{N-1} (x_i - \bar{x})^2}{N-1} < (x_N - x)^2$$

Dividiendo ambos términos por N

$$\frac{\sum_{i=0}^{N-1} (x_i - x)^2}{N(N-1)} < \frac{(x_N - x)^2}{N}$$

Multiplicando por $(N-(N-1))$

$$\begin{aligned} \frac{(N - (N-1)) \sum_{i=0}^{N-1} (x_i - x)^2}{N(N-1)} &< \frac{(x_N - x)^2}{N} \\ \frac{N \sum_{i=0}^{N-1} (x_i - x)^2}{N(N-1)} - \frac{(N-1) \sum_{i=0}^{N-1} (x_i - x)^2}{N(N-1)} &< \frac{(x_N - x)^2}{N} \\ \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})^2}{(N-1)} &< \frac{(x_N - x)^2}{N} + \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})^2}{N} \end{aligned}$$

Con lo que finalmente queda

$$\frac{\sum_{i=0}^{N-1} (x_i - x)^2}{(N-1)} < \frac{\sum_{i=0}^N (x_i - x)^2}{N} \quad (7-4)$$

La desigualdad presentada en (7-4) indica que la varianza cuando se tienen $N-1$ muestras es menor que cuando se tienen N muestras (tomando en cuenta que existe un

desplazamiento constante y en la misma dirección entre muestra y muestra). Esto indica que conforme se aumenta la cantidad de muestras, la varianza crecerá.

7.3 Invertibilidad de la matriz B

Se tiene la expresión

$$X(t_k) - \bar{X} = A(X(t_{k-1}) - \bar{X}) + Bw_k$$

Si se despeja la variable aleatoria, se obtiene

$$B^{-1}(X(t_k) - X - A(X(t_{k-1}) - \bar{X})) = w_k$$

Donde w_k es un vector de n variables aleatorias distribuidas $\mathcal{N}(0,1)$. Si se considera

$BB^T = P$ donde P es la matriz de covarianza, se obtiene $B = \sqrt{P} = P^{1/2}$. Dado que P es una matriz definida positiva, existe una descomposición tal que

$$P = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{n1} \\ \vdots & \ddots & \vdots \\ u_{1n} & \cdots & u_{nn} \end{bmatrix}$$

Donde los $[u_{a0} \dots u_{an}]$ son los auto-vectores de P y λ_i son los autovalores de P^{74} . Si se considera

$$B = P^{1/2} = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{n1} \\ \vdots & \ddots & \vdots \\ u_{1n} & \cdots & u_{nn} \end{bmatrix}$$

y se toma la inversa se obtiene

$$B^{-1} = P^{-1/2} = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} \frac{1}{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\lambda_n} \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{n1} \\ \vdots & \ddots & \vdots \\ u_{1n} & \cdots & u_{nn} \end{bmatrix}$$

Con lo cual se puede ver que B^{-1} es invertible.

⁷⁴ Los autovalores son todos positivos y mayores que cero, por ser la matriz P definida positiva.

7.4 Condiciones sobre β para alcanzar el estado de equilibrio

Se tiene que $P(t_k) = AP(t_{k-1})A^T + BB^T$ (en el caso auto-regresivo de segundo orden el principio es el mismo). Esto mismo se puede expresar como

$$P(t_k) = A^{k+1}P(t_0)(A^T)^{k+1} + \sum_{i=0}^k A^i BB^T A^i, \text{ lo cual nos da una fórmula cerrada más cómoda}$$

para trabajar. Para que esta fórmula cerrada converja es necesario que A sea convergente, y esto sucede si $\rho(A) < 1$. Haciendo la consideración de que $\beta = 0$, la matriz A definida en la sección 5.5, quedará de la siguiente forma (siempre considerada en el espacio de formas planar afín):

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) & 0 \end{bmatrix}$$

Para calcular $\rho(A)$, lo primero que se hace es construir el polinomio característico de la matriz. Dicho polinomio se expresa como $\det(A - \lambda I) = 0$. Calculando $A - \lambda I$, se obtiene

$$A - \lambda I = \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) - \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) - \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) - \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) - \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) - \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2\cos(\pi f\tau) - \lambda & 0 \end{bmatrix}$$

Calcular el $\det(A - \lambda I)$ es simple dado que esta matriz es esparza y se puede expresar como bloques de matrices diagonales. De esta forma, el polinomio característico de $A - \lambda I$ queda como

$$\lambda^6 \cdot (2\cos(2\pi f\tau) - \lambda)^6 - 1 = 0$$

$$\lambda^6 \cdot (2 \cos(2\pi f\tau) - \lambda)^6 = 1$$

$$\lambda \cdot (2 \cos(2\pi f\tau) - \lambda) = 1$$

$$2\lambda \cos(2\pi f\tau) - \lambda^2 - 1 = 0$$

Finalmente queda la siguiente ecuación cuadrática

$$-\lambda^2 + 2\lambda \cos(2\pi f\tau) - 1 = 0$$

Para resolverla, se emplea la fórmula para ecuaciones de segundo grado

$$\frac{-2 \cos(2\pi f\tau) \pm \sqrt{4 \cos^2(2\pi f\tau) - 4(-1)(-1)}}{-2} = \frac{-2 \cos(2\pi f\tau) \pm \sqrt{4 \cos^2(2\pi f\tau) - 4}}{-2}$$

Si se observa el discriminante de la ecuación, se pueden observar dos casos

Caso $2\pi f\tau = 0$

$$\frac{-2 \cos(0) \pm \sqrt{4 \cos^2(0) - 4(-1)(-1)}}{-2} = \frac{-2 \pm 0}{-2} = 1$$

Con lo cual $\rho(A) = 1$, haciendo que la matriz A no sea convergente en este caso.

Caso $2\pi f\tau \neq 0$

$$\begin{aligned} \frac{-2 \cos(2\pi f\tau) \pm \sqrt{4 \cos^2(2\pi f\tau) - 4}}{-2} &= \frac{-2 \cos(2\pi f\tau) \pm \sqrt{4(\cos^2(2\pi f\tau) - 1)}}{-2} = \\ \frac{-2 \cos(2\pi f\tau) \pm 2 \sqrt{\cos^2(2\pi f\tau) - 1}}{-2} &= \frac{-2 \cos(2\pi f\tau) \pm 2 \sqrt{-\sin^2(2\pi f\tau)}}{-2} = \\ \frac{-2 \cos(2\pi f\tau) \pm 2 \sin(2\pi f\tau) \cdot (-1)}{-2} &= \frac{-2 \cos(2\pi f\tau) \pm 2 \sin(2\pi f\tau) i}{-2} = \frac{2(\cos(2\pi f\tau) \pm \sin(2\pi f\tau) i)}{2} = \\ \cos(2\pi f\tau) \pm \sin(2\pi f\tau) i & \end{aligned}$$

Si se calcula el módulo de este valor para calcular el radio espectral se obtiene

$$\sqrt{\cos^2(2\pi f\tau) + (\pm \sin(2\pi f\tau))^2} = \sqrt{\cos^2(2\pi f\tau) + \sin^2(2\pi f\tau)} = 1 = 1$$

Con lo cual se puede ver que $\rho(A) = 1$ también en este caso. Por lo tanto la matriz A no es convergente cuando $\beta = 0$, y la ecuación $P(t_k) = AP(t_{k-1})A^T + BB^T$ también será divergente no alcanzándose nunca un estado de equilibrio.

Ahora lo que queda por mostrar, es que cuando $\beta > 0$ la matriz es convergente. Para hacer esto, se empleará el mismo método que en el caso anterior y se mostrará que $\rho(A) < 1$. La matriz $A - \lambda I$, que queda cuando $\beta > 0$, es

$$A - \lambda I = \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -e^{-12\beta\tau} & 0 & 0 & 0 & 0 & 0 & 2e^{-1\beta\tau} \cos(2\pi f\tau) - \lambda \end{bmatrix}$$

En este caso el polinomio característico es igual a

$$\lambda^6 (2e^{-(\beta\tau)} \cos(2\pi f\tau) - \lambda)^6 - e^{-(12\beta\tau)} = 0$$

$$\lambda (2e^{-(\beta\tau)} \cos(2\pi f\tau) - \lambda) = e^{-(2\beta\tau)}$$

Donde queda la siguiente ecuación cuadrática

$$-\lambda^2 + 2\lambda e^{-(\beta\tau)} \cos(2\pi f\tau) - e^{-(2\beta\tau)} = 0$$

Para resolverla se emplea la fórmula de ecuaciones de segundo grado, y se quiere ver que el resultado es siempre menor que 1.

$$\frac{-2e^{-(\beta\tau)} \cos(2\pi f\tau) \pm \sqrt{4e^{-(2\beta\tau)} \cos^2(2\pi f\tau) - 4(-1)(-e^{-(2\beta\tau)})}}{-2} < 1$$

$$\frac{-2e^{-(\beta\tau)} \cos(2\pi f\tau) \pm 2e^{-(\beta\tau)} \sqrt{\cos^2(2\pi f\tau) - 1}}{-2} < 1$$

Aquí se considerarán dos casos:

Caso $\cos^2(2\pi f\tau) = 1$

$$\frac{-2e^{-(\beta\tau)} \pm 2e^{-(\beta\tau)} \cdot 0}{-2} = \frac{-2e^{-(\beta\tau)}}{-2} = e^{-(\beta\tau)} < 1$$

$$\ln e^{-(\beta\tau)} < \ln 1 = 0$$

$$\beta\tau > 0$$

Dado que τ , el intervalo de muestreo es un valor siempre positivo y $\beta > 0$ se cumple que el valor de $\lambda < 1$. Con esto, el caso queda probado.

Caso $\cos^2(2\pi f\tau) < 1$

$$\frac{-2e^{-(\beta\tau)} \cos(2\pi f\tau) \pm 2e^{-(\beta\tau)} \sqrt{\cos^2(2\pi f\tau) - 1}}{-2} = \frac{-2e^{-(\beta\tau)} \cos(2\pi f\tau) \pm 2e^{-(\beta\tau)} \sqrt{-\sin^2(2\pi f\tau)}}{-2} < 1$$

$$e^{-(\beta\tau)} \cos(2\pi f\tau) \pm e^{-(\beta\tau)} \sin(2\pi f\tau) i < 1$$

$$|\cos(2\pi f\tau) \pm \sin(2\pi f\tau) i| < \left| \frac{1}{e^{-(\beta\tau)}} \right|$$

$$1 < \frac{1}{e^{-(\beta\tau)}} \Rightarrow e^{-(\beta\tau)} < 1$$

$$\beta\tau > 0 \Rightarrow \beta > 0$$

Por lo mismo que
en el caso anterior

Se ha mostrado entonces, que con $\beta > 0$, el radio espectral de A es menor que la unidad, lo que equivale a decir que la matriz es convergente. Estos valores de β , son los únicos que permiten alcanzar un estado de equilibrio.

7.5 Demostración del origen del algoritmo de aprendizaje automático de parámetros

En el espacio unidimensional, la expresión de los momentos auto regresivos de segundo orden se escribe como

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + b_0 w_k$$

Donde $b_0 w_k$ es una variable aleatoria tal que las componentes de w_k son variables aleatorias $\mathcal{N}(0,1)$ y $b_0 \neq 0$. La distribución de la variable aleatoria $Y = b_0 w_k$ es

$$F_Y(y) = P(Y \leq y) = P(b_0 w_k \leq y) = P\left(w_k \leq \frac{y}{b_0}\right) = F_{w_k}\left(\frac{y}{b_0}\right)$$

Derivando, se obtiene la función de densidad

$$f_Y(y) = f_{w_k}\left(\frac{y}{b_0}\right) \frac{1}{b_0}$$

Debido a que las componentes de w_k son $\mathcal{N}(0,1)$, se tiene entonces que

$$f_{w_k}(w) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}w^2}$$

De esta forma

$$f_Y(y) = f_{w_k} \left(\frac{y}{b_0} \right) \frac{1}{b_0} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{y^2}{b_0^2}} \frac{1}{b_0} = \frac{1}{\sqrt{2\pi} b_0} e^{-\frac{1}{2} \frac{y^2}{b_0^2}}$$

De lo que finalmente se concluye que $Y = b_0 w_k \sim \mathcal{N}(0, b_0)$.

Se tiene entonces que $P_{b_0 w_k}(w) = P_{x_k - a_1 x_{k-1} - a_2 x_{k-2}}(w)$ con lo que

$$P(x_1 \dots x_k | a_1, a_2, b_0) \propto \prod_k f_Y(y_k) = \prod_k P_{b_0 w_k}(x_k - a_2 x_{k-2} - a_1 x_{k-1})$$

Si se toma el logaritmo

$$L = \log \prod_k P_{b_0 w_k}(x_k - a_2 x_{k-2} - a_1 x_{k-1}) = \sum_{k=3}^M \log P_{b_0 w_k}(x_k - a_2 x_{k-2} - a_1 x_{k-1})$$

Usando que $f_{b_0 w_k}(w) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{w^2}{b_0^2}}$

$$\log \prod_k P_{b_0 w_k}(x_k - a_2 x_{k-2} - a_1 x_{k-1}) = \sum_{k=3}^M \log \frac{1}{\sqrt{2\pi} b_0} e^{-\frac{1}{2} \frac{(x_k - a_2 x_{k-2} - a_1 x_{k-1})^2}{b_0^2}} =$$

Distribuyendo el logaritmo

$$\begin{aligned} &= \sum_{k=3}^M \log \frac{1}{\sqrt{2\pi} b_0} - \frac{1}{2} \frac{(x_k - a_2 x_{k-2} - a_1 x_{k-1})^2}{b_0^2} = \\ &= (M-2) \log \frac{1}{\sqrt{2\pi} b_0} - \frac{1}{2 b_0^2} \sum_{k=3}^M (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2 \end{aligned}$$

Para sacar los valores de a_1 , a_2 y b_0 que maximizan la función de probabilidad logarítmica, se deriva con respecto a cada uno de ellos. De esta forma se obtiene

$$\frac{\partial L}{\partial a_1} = -\frac{1}{b_0^2} \sum_{k=3}^M (-x_{k-1})(x_k - a_2 x_{k-2} - a_1 x_{k-1}) = 0 \quad (7-5)$$

$$\frac{\partial L}{\partial a_2} = -\frac{1}{b_0^2} \sum_{k=3}^M (-x_{k-2})(x_k - a_2 x_{k-2} - a_1 x_{k-1}) = 0 \quad (7-6)$$

$$\frac{\partial L}{\partial b_0} = -\frac{M-2}{b_0} + \frac{1}{b_0^3} \sum_{k=3}^M (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2 = 0 \quad (7-7)$$

Si se distribuye la sumatoria en (7-5) y (7-6) se obtiene

$$-a_1 \sum_{k=3}^M x_{k-1}^2 - a_2 \sum_{k=3}^M x_{k-1} x_{k-2} + \sum_{k=3}^M x_{k-1} x_k = 0$$

$$-a_1 \sum_{k=3}^M x_{k-2} x_{k-1} - a_2 \sum_{k=3}^M x_{k-2}^2 + \sum_{k=3}^M x_{k-2} x_k = 0$$

Si se llama $r_{ij} = \sum_{k=3}^M x_{k-i} x_{k-j}$ $i, j = 0, 1, 2$, se puede obtener el siguiente sistema de ecuaciones

$$\begin{cases} -\hat{a}_1 r_{11} - \hat{a}_2 r_{12} + r_{10} = 0 \\ -\hat{a}_1 r_{21} - \hat{a}_2 r_{22} + r_{20} = 0 \end{cases}$$

Que es el sistema que resuelve en el paso 2 del algoritmo de la Figura 5-5.

Por último se considerará (7-7). Si se saca factor común $\frac{1}{b_0^3}$

$$-(M-2)b_0^2 + \sum_{k=3}^M (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2 = 0 \quad (7-8)$$

Desarrollando el término $\sum_{k=3}^M (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2$ se obtiene

$$\begin{aligned} (x_k - a_2 x_{k-2} - a_1 x_{k-1})^2 &= x_k^2 - 2(a_2 x_{k-2} + a_1 x_{k-1})x_k + (a_2 x_{k-2} + a_1 x_{k-1})^2 = \\ &= x_k^2 - 2a_2 x_{k-2} x_k - 2a_1 x_{k-1} x_k + a_2^2 x_{k-2}^2 + 2a_1 a_2 x_{k-1} x_{k-2} + a_1^2 x_{k-1}^2 \end{aligned}$$

Se toma ahora la sumatoria

$$= \sum_{k=3}^M x_k^2 + a_2^2 \sum_{k=3}^M x_{k-2}^2 + a_1^2 \sum_{k=3}^M x_{k-1}^2 + 2 \left[(-a_2) \sum_{k=3}^M x_{k-2} x_k - a_1 \sum_{k=3}^M x_{k-1} x_k + a_1 a_2 \sum_{k=3}^M x_{k-1} x_{k-2} \right] =$$

Empleando $r_{ij} = \sum_{k=3}^M x_{k-i} x_{k-j}$ $i, j = 0, 1, 2$, $a_2 r_{22} = r_{20} - a_1 r_{21}$ y $a_1 r_{11} = r_{10} - a_2 r_{12}$, se reemplaza y se obtiene

$$\begin{aligned} &= r_{00} + a_2^2 r_{22} + a_1^2 r_{11} + 2[(-a_2)r_{20} - a_1 r_{10} + a_1 a_2 r_{12}] = \\ &= r_{00} + a_2(r_{20} - a_1 r_{21}) + a_1(r_{10} - a_2 r_{12}) + 2[(-a_2)r_{20} - a_1 r_{10} + a_1 a_2 r_{12}] = \\ &= r_{00} + a_2 r_{20} - a_2 a_1 r_{21} + a_1 r_{10} - a_1 a_2 r_{12} - 2a_2 r_{20} - 2a_1 r_{10} + 2a_1 a_2 r_{12} = \end{aligned}$$

Dado que r_{ij} representa una correlación entre valores, se cumple que $r_{ij} = r_{ji}$. De esta manera,

$$= r_{00} + a_2 r_{20} - a_2 a_1 r_{21} + a_1 r_{10} - a_1 a_2 r_{12} - 2a_2 r_{20} - 2a_1 r_{10} + 2a_1 a_2 r_{12} = r_{00} - a_2 r_{20} - a_1 r_{10}$$

Reemplazando en (7-8), finalmente queda

$$\hat{b}_0^2 = \frac{1}{M-2} (r_{00} - \hat{a}_2 r_{02} - \hat{a}_1 r_{10})$$

Que es lo que se resuelve en el paso 3 del algoritmo de la Figura 5-5

Por último, el caso multidimensional es una generalización del caso unidimensional. Para una demostración referirse a [5].



Capítulo 8: Referencias

- [1] Baumberg A., Hogg, D. *Learning flexible models from image sequences*. In Eklundh, J.-O., editor, Proc. 3rd European Conf. Computer Vision. Springer-Verlag (1994).
- [2] Baumberg A., Hogg D. *Using Shape and Intensity to Track Non-Rigid Objects*. School of Computer Studies Research Report Series. Report 94.14 (1994).
- [3] Baumberg A., Hogg D. *An Efficient Method for Contour Tracking using Active Shape Models*. School of Computer Studies Research Report Series. Report 94.11 (1994).
- [4] Baumberg A., Hogg D. *Learning Deformable Models for Tracking Human Motion*. Phd Thesis, The University of Leeds, School of Computer Studies (1995).
- [5] Blake A., Isard M. *Active Contours*. Ed. Springer (1998).
- [6] Blake A., Isard M., Reynard D. *Learning to track the visual motion of coutours*. Department of Engineering Science, University of Oxford, Oxford, UK (1995).
- [7] Blake A., Yuille A. *Active Vision*. Editors (1992).
- [8] Blake A., Bascle B., Isard M., MacCormick J. *Statistical models of visual shape and motion*. Department of Engineering Science, University of Oxford, Oxford, UK (1998).
- [9] Blake A., Isard M. *3D position, attitude and shape input using video tracking of hands and lips*. Proc. ACM Siggraph, 185-192 (1994).
- [10] Blake A., Isard M. *Contour tracking by stochastic propagation of conditional density*. Proceedings of European Conference of Computer Vision, pp. 343-356, Cambridge, UK (1996)
- [11] Boudier T. *Developing a deformation model for complex-shaped contours*. BioInformatique Structurale, Paris Cedex, France.
- [12] Castells, V., Kimmel R., Sapiro G. *Geodesic Active Contours*. International Journal of Computer Vision 22(1), Kluwer Academic Publishers, pp. 61-79, The Netherlands (1997).
- [13] Curwen, R. *Dynamic and Adaptive Contours*. PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, UK (1993).
- [14] DeCarlo D., Metaxas D. *Adjusting Shape Parameters using Model-Based Optical Flow Residuals*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 24, Issue 6, pp. 814-823 (2002).
- [15] Freedman D. *Active Contours for Tracking Distributions*. Computer Science Department Rensselaer Polytechnic Institute (2002).
- [16] Johnson N., Hogg D. *Learning the Distribution of Object Trajectories for Event*

- Recognition*. University of Leeds, United Kingdom, (1996).
- [17] Kaas M., Witkin A., Terzopoulos D. *Snakes: Active Contours Models*. International Journal of Computer Vision, Kluwer Academic Publishers, Boston, pp. 321-331 (1988).
 - [18] Khan S., Shah M. *Tracking People in Presence of Occlusion*. Computer Vision Lab, School of Computer Science, University of Central Florida (2000).
 - [19] Magee D. *Tracking Multiple Vehicles using Foreground, Background and Motion Models*. University of Leeds, School of Computer Studies, Research Report Series (2001).
 - [20] Nickels K., Hutchinson S. *Estimating Uncertainty in SSD-Based Feature Tracking*. Image and Vision Computing, Vol. 20, pp. 47-58 (2002).
 - [21] Nickels K., Hutchinson S. *Model-Based Tracking of Complex Articulated Objects*. IEEE Transactions on Robotics and Automation, Vol 17. NO. 1 (2002).
 - [22] Paragios N., Deriche R. *Coupled Geodesic Active Regions for Image Segmentation*. INRIA (1999)
 - [23] Paragios N., Deriche R. *Geodesic Active Regions and level sets for the detection and tracking of moving objects*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 3 (2000)
 - [24] Paragios N., Deriche R. *Geodesic Active Regions for Motion Estimation and Tracking*. Rapport de Recherche, INRIA, No. 3631 (1999).
 - [25] Paragios N., Deriche R. *Geodesic Active Regions for Texture Segmentation*. Rapport de Recherche, INRIA, No. 3440 (1998).
 - [26] Paragios N., Deriche R. *Geodesic Active Regions for Supervised Texture Segmentation*. Proceedings of the 7th IEEE International Conference in Computer Vision, Greece (1999).
 - [27] Reynard D., Wildenberg A., Blake A., Marchant J. *Learning Dynamics of Complex Motions from Image Sequences*. Proceedings of the European Conference of Computer Vision, pp. 357-368 Cambridge, UK (1996).
 - [28] Sonka M., Hlavac V. and Boyle R. *Image Processing, Analysis, and Machine Vision*. ITP (1998).
 - [29] Toresani L., Yang D., Alexander E., Bregler C. *Tracking and Modelling Non-Rigid Objects with Rank Constraints*. Standford University, Standford (2001).
 - [30] Westin C.F., Lorigo L., Faugeras O., Grimson E., Dawson S., Norbash A. and Kikinis R. *Segmentation by Adaptive Geodesic Active Contours*. In Proceedings of Third International Conference On Medical Robotics, Imaging and Computer Assisted Surgery, Pittsburgh, Pennsylvania, USA, pp. 266-275 (2000).
 - [31] Wilhelms J., Van Gelder A., Atkinson-Derman L., Luo A. *Human Motion from Active Contours*. Workshop on Human Motion, Austin, Texas, USA, pp. 155-160 (2000).

