

Procesamiento de Información Visual

Tema 4. Análisis del Movimiento

José Manuel Fuertes García
Manuel J. Lucena López

Departamento de Informática
Universidad de Jaén



11 de mayo de 2021

Índice

Procesamiento de vídeo

Desentrelazado

Movimiento

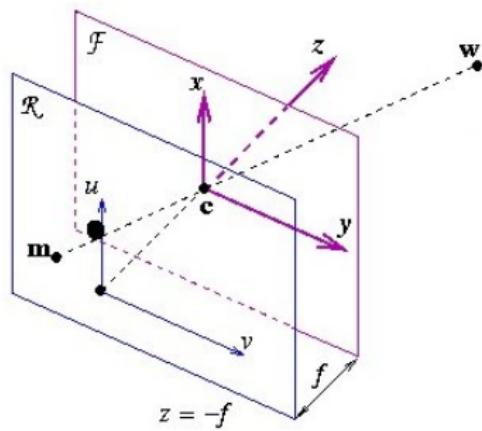
Seguimiento (*Tracking*)

Naturaleza de la información presente en un vídeo

- ▶ Un vídeo es un conjunto de imágenes, normalmente ordenadas en el tiempo.
- ▶ La secuencia puede ser natural o sintética.
- ▶ Puede proceder de una cámara o de otro tipo de dispositivos (ecografías, telemetría, resonancias nucleares, etc).
- ▶ Las técnicas de análisis deben tener en cuenta el proceso de formación de la imagen.
- ▶ El campo del análisis y procesamiento de imágenes y vídeo es enormemente complejo y variado.

La cámara

- ▶ Recoge la luz y la dirige a un sensor.
- ▶ Puede modelizarse matemáticamente (calibración).
- ▶ Una cámara *real* suele producir aberraciones.
- ▶ Proporciona una correspondencia entre puntos del universo y puntos en la imagen.



Modelo abstracto de cámara.

Parámetros

► **Resolución espacial:**

La imagen se divide en $m \times n$ píxeles.

► **Relación de aspecto:**

Proporción del alto con respecto del ancho de la imagen.

► **Resolución temporal:**

Frecuencia de muestreo (cuadros por segundo).

► **Cuantizado:**

Nº de bits que se dedican a cada píxel.

Representación de imágenes

- ▶ Espacial / frecuencial (*Fourier, Wavelets, . . .*).
- ▶ Piramidal (*Gaussiana o Laplaciana*).

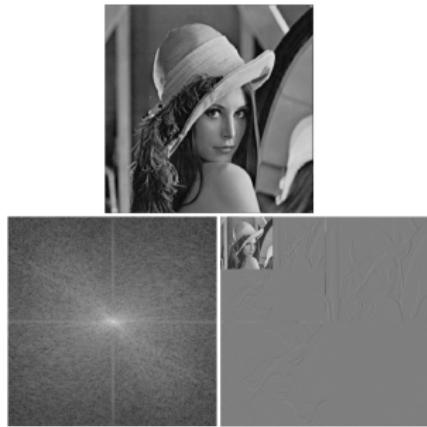
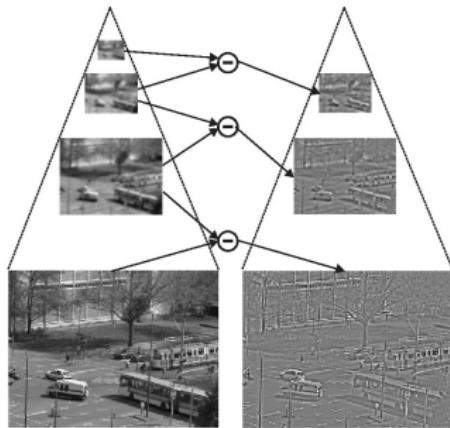


Imagen espacial y frecuenciales asociadas.



Pirámides Gaussiana y Laplaciana.

Tipos de secuencias de imágenes

- ▶ Coordenadas euclídeas / (log)Polares.
- ▶ Monobanda / Multibanda.
- ▶ Entrelazada / Progresiva.
- ▶ Monocanal / Multicanal o multivista (ej: secuencias estereoscópicas).



Imagen LogPolar.



Imagen estereoscópica.

Procesamiento en línea/fuera de línea

Procesamiento fuera de línea:

- ▶ El vídeo es considerado como un volumen tridimensional.
- ▶ Disponemos de toda la información, tanto pasada como futura.
- ▶ El tiempo de respuesta no suele ser crítico.

Procesamiento en línea:

- ▶ El vídeo se va leyendo o captando conforme se procesa.
- ▶ Tiempo de respuesta crítico.
- ▶ Sólo disponemos de la información *pasada*, a veces de pocos cuadros.

Índice

Procesamiento de vídeo

Desentrelazado

Movimiento

Seguimiento (*Tracking*)

Vídeo entrelazado

- ▶ Los sistemas antiguos de televisión enviaban por separado los cuadros con líneas pares y los cuadros con líneas impares.
- ▶ Las resoluciones espacial y temporal se *mezclan*.



¿Qué es el vídeo entrelazado?

- ▶ En una secuencia entrelazada, la matriz de puntos que compone la imagen se divide en dos *campos*: uno contiene las líneas pares, y otro las impares.
- ▶ El muestreo de cada campo se hace en diferentes instantes de tiempo:

$$I_{par}(t), I_{impar}(t+1), I_{par}(t+2), I_{impar}(t+3) \dots$$



¿Por qué usamos vídeo entrelazado?

- ▶ **Persistencia de la visión:** La información capturada por la retina permanece un breve intervalo de tiempo.
- ▶ Si pasamos imágenes a una velocidad adecuada, se perciben como una secuencia continua.
- ▶ El vídeo entrelazado permite mantener una tasa de refresco alta con la **mitad del ancho de banda**.
- ▶ Funciona de forma natural en los CRT.
- ▶ Los sistemas de vídeo analógico tradicionales (PAL, SECAM, NTSC...) son entrelazados.

¿En qué consiste el desentrelazado de vídeo?

- ▶ Consiste en convertir una secuencia entrelazada en una secuencia de cuadros completos, cada uno asociado a un instante de tiempo (**vídeo progresivo**).
- ▶ Los dispositivos modernos de visualización (LCD, Plasma...) no permiten vídeo entrelazado de forma natural.
- ▶ Los estándares de vídeo modernos *tienden* al formato progresivo.

Necesitamos el desentrelazado para poder **convertir** contenidos a formatos progresivos, y para **representar** material entrelazado en dispositivos **progresivos**.

Definiciones

Desentrelazado:

- ▶ Entrada → Secuencia de *campos*:

$$\begin{cases} I_{par}(x, y, t) & \text{si } t \bmod 2 = 0 \\ I_{impar}(x, y, t) & \text{en caso contrario.} \end{cases}$$

- ▶ Salida → Secuencia de *cuadros*:

$$I(x, y, t) = \begin{cases} I(x, y, t) & \text{si } y \bmod 2 = t \bmod 2 \\ I_i(x, y, t) & \text{en caso contrario.} \end{cases}$$

- ▶ I_i son **cuadros interpolados**, que tendremos que calcular.

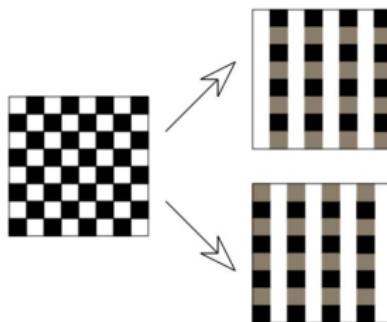
Teorema de muestreo (Nyquist–Shannon)

La **frecuencia de muestreo** debe ser mayor que dos veces el ancho de banda de la señal de entrada, para poder reconstruir la señal original de forma exacta a partir de las muestras.

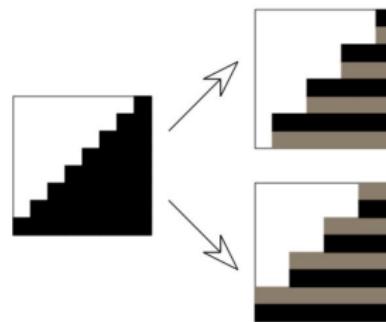
- ▶ Si no se cumple, se producen efectos de **aliasing**.
- ▶ El entrelazado reduce a la mitad la frecuencia de muestreo vertical para cada campo.
- ▶ Si rellenamos los cuadros copiando los campos presentes, aparecen **artefactos**.
- ▶ Es necesario **reconstruir** de alguna manera la información perdida.

Artefactos de desentrelazado en imágenes estáticas

¿Es suficiente con calcular cada cuadro duplicando el campo presente en cada instante?



Vibración horizontal.



Diente de sierra.

Estas imágenes no cumplen el Teorema de Muestreo.

Tipos de algoritmos de desentrelazado

- ▶ No basados en compensación de movimiento
 - ▶ Atendiendo al tipo de interpolación:
 - ▶ Lineales.
 - ▶ No lineales.
 - ▶ Atendiendo a qué elementos se interpolan:
 - ▶ Espaciales (*intra-campo*).
 - ▶ Temporales (*inter-campo*).
 - ▶ Espacio-temporales.
- ▶ Basados en compensación de movimiento
 - ▶ Retroproyección temporal, Recursivo en el tiempo, Adaptativo recursivo...

Desentrelazado lineal espacial "Bob"

- ▶ Explota la **correlación espacial** entre muestras vecinas verticales en un campo.
- ▶ Funciona mejor con secuencias dinámicas.
- ▶ Se calcula promediando las líneas superior e inferior:

$$I_s(x, y, t) = \begin{cases} I(x, y, t) & y \bmod 2 = t \bmod 2 \\ \frac{1}{2} \sum_{k \in \{-1, 1\}} I(x, y + k, t), & \text{en caso contrario.} \end{cases}$$

Desentrelazado lineal temporal "Weave"

- ▶ Explota la **correlación temporal** entre muestras.
- ▶ Funciona mejor con secuencias estáticas.
- ▶ Se calcula insertando (o repitiendo) el campo anterior en el cuadro actual:

$$I_t(x, y, t) = \begin{cases} I(x, y, t) & y \bmod 2 = t \bmod 2 \\ I(x, y, t - 1), & \text{en caso contrario.} \end{cases}$$

Desentrelazado VT

- ▶ Método espacio-temporal lineal.
- ▶ Se calcula con una suma ponderada:

$$I_{VT}(x, y, t) = \begin{cases} I(x, y, t), & y \bmod 2 = t \bmod 2 \\ \sum_{k,m} (x, y + k, t + m) h(k, m), & \text{en caso contrario.} \end{cases}$$

$$18h(k, m) = \begin{cases} 1, 8, 8, 1, & (k = -3, -1, 1, 3) \wedge (m = 0) \\ -5, 10, -5, & (k = -2, 0, 2) \wedge (m = -1) \\ 0, & \text{en caso contrario.} \end{cases}$$

Desentrelazado no lineal espacio-temporal

- ▶ Depende de un factor γ , que mide el *movimiento* en cada zona de la imagen.
- ▶ Es una combinación lineal de I_s e I_t :

$$I_{ts}(x, y, t) = \gamma I_s(x, y, t) + (1 - \gamma) I_t(x, y, t)$$

- ▶ Mucho movimiento, $\gamma \rightarrow 1$, $I_{ts} \rightarrow I_s$.
- ▶ Poco movimiento, $\gamma \rightarrow 0$, $I_{ts} \rightarrow I_t$.
- ▶ El cálculo del factor γ no es trivial.
- ▶ Podemos combinar cualesquiera dos filtros adaptados a secuencias estáticas y dinámicas.

Índice

Procesamiento de vídeo

Desentrelazado

Movimiento

Seguimiento (*Tracking*)

Movimiento (Flujo óptico)

- ▶ Mide el desplazamiento aparente de los patrones de píxeles en el plano de la imagen.
- ▶ Es un problema **mal definido** (*ill-posed*).
- ▶ Dado por la restricción de constancia de brillo:

$$I(x, y, t) \approx I(x + \delta x, y + \delta y, t + \delta t)$$

que supone que los valores de intensidad son aproximadamente iguales en los lugares donde está el punto en dos instantes de tiempo consecutivos. Esto da lugar a la ecuación:

$$I_x u + I_y v + I_t = 0$$

- ▶ I_x, I_y, I_t son las derivadas de la imagen con respecto a las coordenadas x, y y el tiempo t .

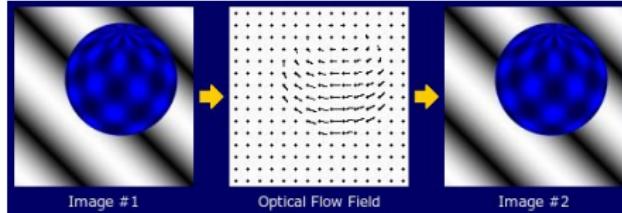
Algunas aplicaciones del flujo óptico

- ▶ Compresión de vídeo.
 - ▶ Para reducir el espacio necesario de almacenamiento, se calcula el desplazamiento de los *macrobloques* y se envía junto con las diferencias entre el cuadro a empaquetar y el resultado de aplicar el movimiento al cuadro anterior.
- ▶ Aumento de la velocidad de muestreo temporal.
 - ▶ Para generar un cuadro intermedio entre otros dos dados, es necesario calcular el flujo óptico entre ambos y aplicar las deformaciones de forma gradual.
- ▶ Análisis de patrones de movimiento para detectar acciones.

Cómo calcular el flujo

- ▶ Las derivadas I_x , I_y y I_t se pueden calcular mediante diferencias o cualquier otro método.
- ▶ La ecuación de constancia de brillo proviene de una expansión de Taylor de primer orden, por lo que solo es válida cuando los incrementos en x , y y t son pequeños.
- ▶ Esto se puede *corregir* si partimos de una aproximación del flujo (\hat{u} , \hat{v}) cercana al valor real, y las diferencias en t las calculamos con los valores de intensidad en la posición a la que nos lleva la misma.

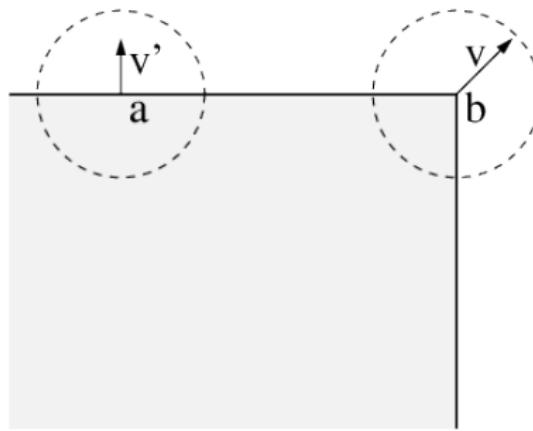
$$I_t = I(x + \hat{u}, y + \hat{v}, t + 1) - I(x, y, t).$$



Flujo Óptico entre dos imágenes sintéticas.

El problema de la apertura

- ▶ Disponemos de una única ecuación lineal en cada punto, con dos incógnitas.
- ▶ El flujo no puede calcularse sin restricciones adicionales.



En *b* podemos calcular el flujo completo, pero en *a* solo la componente en la dirección del gradiente.

Método de Horn y Schunk

- ▶ Propone una condición global de *suavidad* en el campo de flujo:

$$\int \int (I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \, dx \, dy$$

- ▶ El flujo se obtiene mediante optimización, de forma iterativa.
- ▶ Es un modelo poco realista, porque supone que el flujo no posee discontinuidades.
- ▶ Existen métodos que calculan el flujo de forma bidireccional para detectar las discontinuidades.

Método de Lucas y Kanade

- ▶ Emplea una condición *local*, suponiendo el flujo constante en entornos locales pequeños.
- ▶ La solución (u, v) viene dada por:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

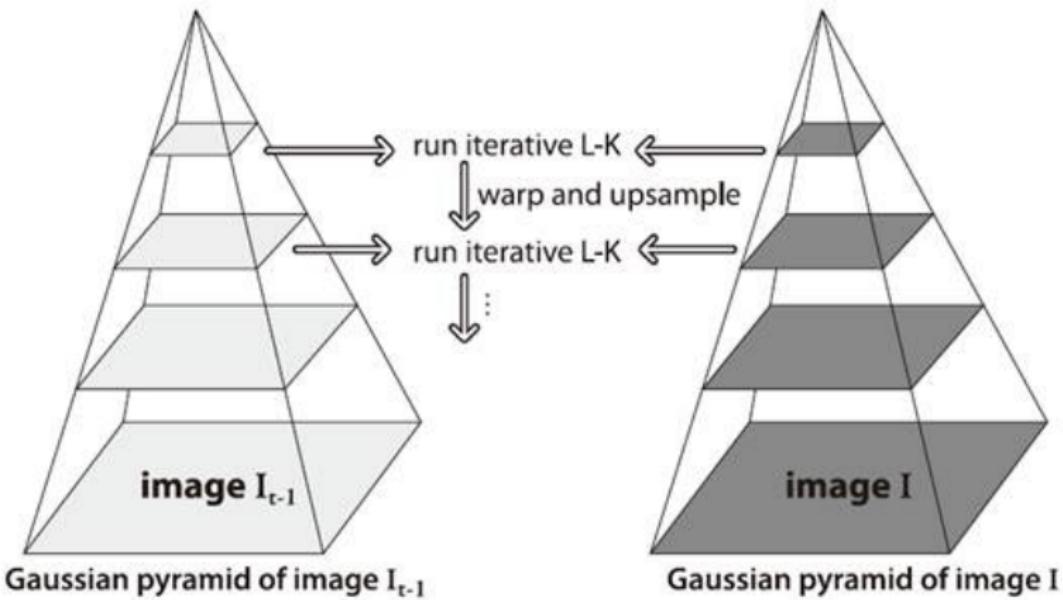
- ▶ Existen modelos más recientes que suponen que el flujo no es constante en pequeños entornos, pero sí parametrizable.

Procesamiento jerárquico

- ▶ La ecuación de flujo funciona si se toma como partida un vector próximo al valor real de flujo.
- ▶ Esto no se cumple para valores de flujo grandes.
- ▶ Empleando una pirámide gaussiana de imágenes, se calcula el flujo para las de un nivel, y se usa como punto de partida para el siguiente.

Estimación jerárquica de flujo óptico

Coarse-to-fine optical flow estimation



Flujo paramétrico

- ▶ Hay situaciones en las que todo el flujo de un par de cuadros corresponde a un mismo movimiento.
- ▶ Estimar el flujo corresponde a estimar los parámetros:
 - ▶ Flujo de traslación: 2 parámetros.
 - ▶ Flujo afín: 6 parámetros.
 - ▶ Movimiento 3D de una superficie planar: 8 parámetros.
 - ▶ Movimiento de la cámara: estimando la profundidad de cada punto en una escena rígida.

Flujo constante

$$u(x, y, t) = a$$

$$v(x, y, t) = d$$

- ▶ Se usa un sistema de ecuaciones similar al de Lucas-Kanade:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} a \\ d \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Flujo afín

$$u(x, y, t) = a + bx + cy$$

$$v(x, y, t) = d + ex + fy$$

$$\mathcal{M} \cdot \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = - \begin{bmatrix} \sum l_x l_t \\ \sum x l_x l_t \\ \sum y l_x l_t \\ \sum l_y l_t \\ \sum x l_y l_t \\ \sum y l_y l_t \end{bmatrix}$$

Flujo afín

$$\mathcal{M} = \begin{bmatrix} \sum I_x^2 & \sum xI_x^2 & \sum yI_y^2 & \sum I_xI_y & \sum xI_xI_y & \sum yI_xI_y \\ \sum xI_x^2 & \sum x^2I_x^2 & \sum xyI_y^2 & \sum xI_xI_y & \sum x^2I_xI_y & \sum xyI_xI_y \\ \sum yI_x^2 & \sum xyI_x^2 & \sum y^2I_x^2 & \sum yI_xI_y & \sum xyI_xI_y & \sum y^2I_xI_y \\ \sum I_xI_y & \sum xI_xI_y & \sum yI_xI_y & \sum I_y^2 & \sum xI_y^2 & \sum yI_y^2 \\ \sum xI_xI_y & \sum x^2I_xI_y & \sum xyI_xI_y & \sum xI_y^2 & \sum x^2I_y^2 & \sum xyI_y^2 \\ \sum yI_xI_y & \sum xyI_xI_y & \sum y^2I_xI_y & \sum yI_y^2 & \sum xyI_y^2 & \sum y^2I_y^2 \end{bmatrix}$$

Flujo planar

- ▶ También llamado transformación 2D pseudo-proyectiva.

$$u(x, y, t) = a + bx + cy + gx^2 + hxy$$

$$v(x, y, t) = d + ex + fy + gxy + hy^2$$

Flujo de un par estéreo

- ▶ El flujo, también llamado *disparidad*, no tiene componente vertical ($v(x, y) = 0$).
- ▶ No existe el problema de la apertura, ya que tenemos una ecuación y una incógnita en cada píxel. Siempre que la imagen no sea constante en entornos locales, se puede estimar la disparidad, aunque puede haber ambigüedades.
- ▶ La magnitud de la componente horizontal del flujo es inversamente proporcional a la distancia del objeto a la cámara.
- ▶ Esto permite estimar la estructura tridimensional de la escena.

Flujo de un par estéreo

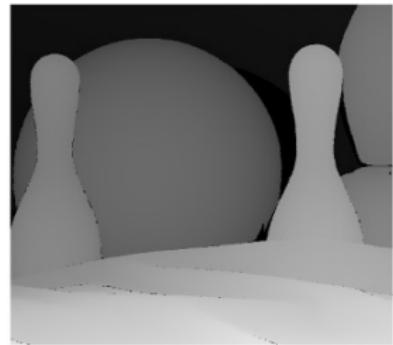
Left Image



Right Image

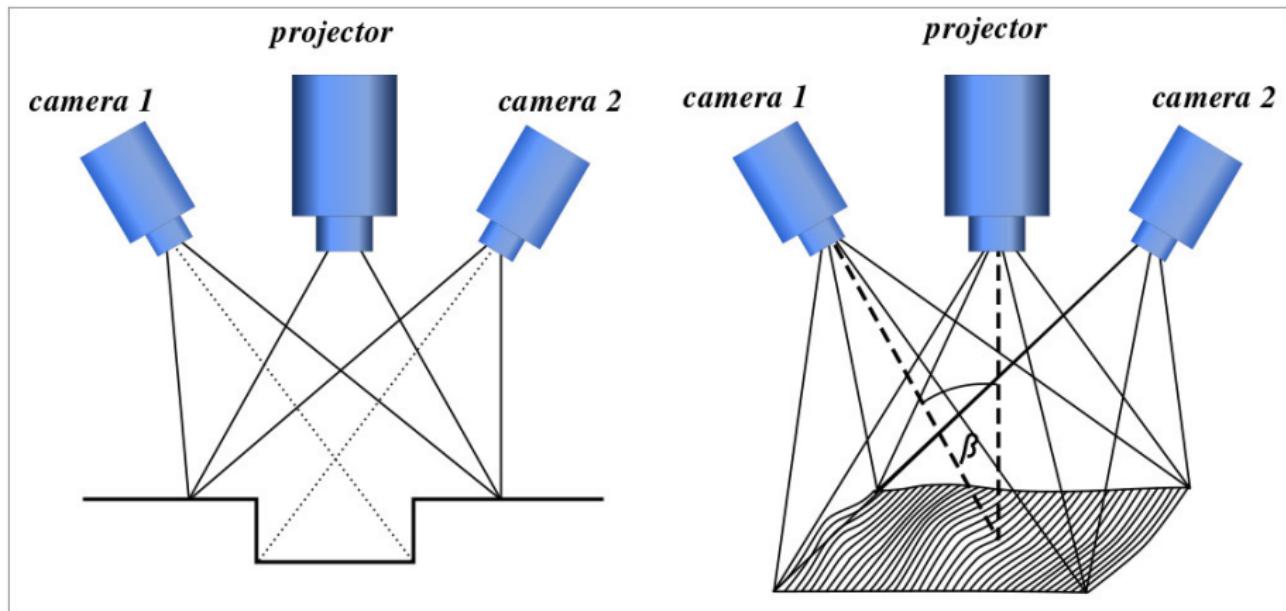


Disparity Map (left)

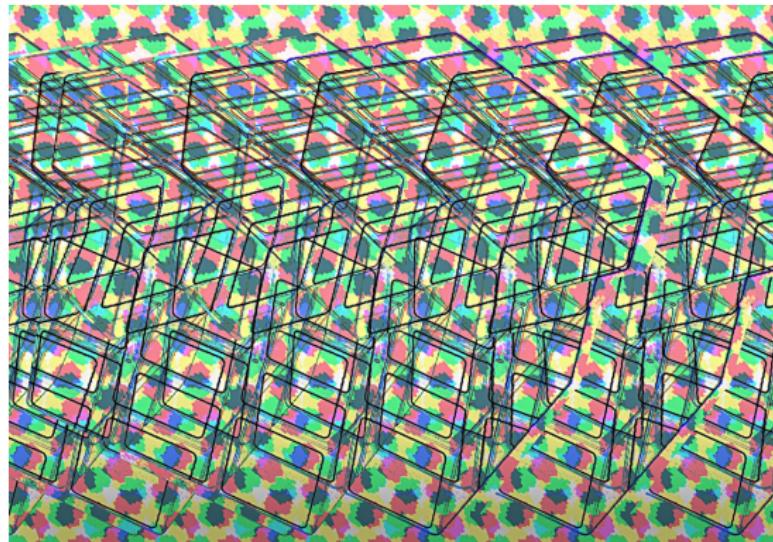


Escáner 3D

- ▶ Se proyecta un patrón sobre el objeto, que permite analizar la disparidad entre dos cámaras sin ambigüedades.



Estereograma de punto aleatorio



Reconstrucción de una escena a partir de múltiples vistas

- ▶ Partimos de una escena estática, por lo que podemos suponer que se trata de una escena *rígida*, y que lo que se ha movido es la cámara.
- ▶ Empleando características tipo SIFT se establecen correspondencias entre las imágenes.
- ▶ Mediante optimización, se asigna a cada punto una posición 3D compatible con las medidas.

Reconstrucción de una escena a partir de múltiples vistas



Índice

Procesamiento de vídeo

Desentrelazado

Movimiento

Seguimiento (*Tracking*)

Tracking

Para hacer seguimiento (*tracking*), explotaremos la *coherencia temporal* de una secuencia de vídeo.

Coherencia temporal: Un cuadro presenta normalmente pocas diferencias con el anterior.

- ▶ Si tenemos localizado un objeto en un cuadro, es lógico buscarlo cerca en el siguiente cuadro.
- ▶ A partir de un modelo de la dinámica del objeto, podemos *predecir* su posición probable en el siguiente cuadro.

Ventajas:

- ▶ Reducimos el ámbito de búsqueda, y el coste computacional.
- ▶ Si hay varios objetos parecidos, podemos discriminarlos.

Notación

- ▶ X_t : Estado del objeto en el instante t .
 - ▶ Vector que modeliza la configuración concreta del objeto (posición, escala, deformación, etc.).
- ▶ Z_t : Variable aleatoria que representa las medidas que pueden obtenerse de la imagen.
 - ▶ Pueden ser fronteras, flujo óptico, correlación de patrones, etc.
 - ▶ Podemos emplear medidas del mismo o diferentes tipos.
- ▶ $Z_t = z_t$: Medida concreta obtenida en el instante t .

Fusión temporal de información para seguimiento

Para hacer seguimiento, hemos de integrar dos fuentes de información (*fusión temporal*):

- ▶ Información de la *historia* pasada del objeto, que permite calcular su evolución probable.
- ▶ Evidencias presentes en la imagen en el instante actual (Z_t).

Este proceso puede dividirse en tres fases:

- ▶ Predicción.
- ▶ Asociación de datos.
- ▶ Corrección.

Fusión temporal de información para seguimiento

Predicción:

- ▶ Sabiendo que hemos medido z_0, \dots, z_{t-1} , ¿cuál es la configuración esperada para X , en el instante t ?
- ▶ Modelizaremos la distribución:

$$p(X_t | Z_0 = z_0, \dots, Z_{t-1} = z_{t-1})$$

Fusión temporal de información para seguimiento

Asociación de datos:

- ▶ Debemos seleccionar, de todos los posibles z_t presentes en la imagen, cuáles son relevantes para el seguimiento.
- ▶ Partiremos de las medidas z_0, \dots, z_{t-1} utilizadas en instantes anteriores.

Fusión temporal de información para seguimiento

Corrección:

- ▶ Utilizando el valor z_t , medido en el instante actual, adaptamos la distribución de probabilidad sobre X .
- ▶ En este caso, modelizaremos:

$$p(X_t | Z_0 = z_0, \dots, Z_t = z_t)$$

Asunciones de independencia

Para simplificar el problema, asumiremos que:

- ▶ La configuración X_t del objeto solo está afectada por la configuración en el instante anterior (Markov):

$$p(X_t | X_0, \dots, X_{t-1}) = p(X_t | X_{t-1})$$

- ▶ Las distintas medidas obtenidas en el instante t son independientes tanto entre ellas como de las medidas en instantes anteriores:

$$p(Z_0 = z_0, \dots, Z_t = z_t | X_t) = \prod_{i=0}^t p(Z_i = z_i | X_i)$$

Proceso de seguimiento

- ▶ En cada cuadro, partimos de una distribución *a priori* $p(X)$ sobre las posibles configuraciones del objeto.
- ▶ Queremos estimar la distribución *a posteriori* $p(X|Z)$.
- ▶ Aplicando el Teorema de Bayes:

$$p(X|Z) \propto p(X) \cdot p(Z|X)$$

- ▶ $p(Z|X)$ es lo que se conoce como *modelo de observación* o *función de verosimilitud*.

Modelo dinámico

Define, a partir de la *historia* del objeto, la configuración esperada $p(X)$ para un instante de tiempo.

- ▶ No sólo representa desplazamientos, sino también deformaciones.
- ▶ Optaremos por un modelo *generativo*, definiendo el nuevo valor de X a partir de los anteriores, más una componente aleatoria.

Modelos de primer orden

Modelo autorregresivo de primer orden:

$$X_t - \bar{X} = A(X_{t-1} - \bar{X}) + Bw_k$$

siendo w_k un vector de variables aleatorias $\mathcal{N}(0, 1)$.

- ▶ Permite representar movimientos direccionales con una velocidad media constante.
- ▶ Ejemplo: **Movimiento Browniano**:

$$X_t = X_{t-1} + Bw_k$$

Modelos de segundo orden

Modelo autorregresivo de segundo orden:

$$X_t - \bar{X} = A_2(X_{t-2} - \bar{X}) + A_1(X_{t-1} - \bar{X}) + Bw_k$$

- ▶ Permite representar movimientos armónicos.
- ▶ Emplea los dos estados anteriores en el tiempo.
- ▶ Puede recuperarse la asunción de Markov agrupando los dos estados:

$$\mathcal{X}_t = \begin{pmatrix} X_{t-1} \\ X_{t-2} \end{pmatrix}$$

El filtro de Kalman

Supongamos que queremos estimar un proceso que evoluciona con el tiempo, y sobre el que podemos efectuar medidas:

- ▶ Dinámica perfecta → No necesitamos medir nada.
- ▶ Medidas perfectas → No necesitamos modelizar la dinámica.
- ▶ Caso intermedio: **dinámica imprecisa y medidas ruidosas.**
- ▶ Solución: Integrar las dos ⇒ **Filtro de Kalman.**

Características del filtro de Kalman

Es un algoritmo de **proceso de datos óptimo y recursivo**, para estimar el estado de un proceso estocástico lineal.

- ▶ **Proceso de datos:** Maneja información tomada en puntos discretos de tiempo.
- ▶ **Óptimo:** Los resultados minimizan el error cuadrático, cuando se cumplen las condiciones previas.
- ▶ **Recursivo:** No necesita almacenar toda la información anterior en el tiempo cada vez que se toma una nueva medida.

Utilidad del filtro de Kalman

- ▶ Propuesto por R. E. Kalman en 1960.
- ▶ Mecanismo natural para hacer **fusión temporal** cuando las distribuciones involucradas son gaussianas.
- ▶ Permite calcular la evolución de la función de densidad del estado del objeto, $p(X)$.
- ▶ Supone que las medidas empleadas para actualizar el modelo son ruidosas.
- ▶ Se aplica habitualmente en múltiples campos científicos.

Funcionamiento del filtro de Kalman

El filtro de Kalman mantiene una distribución gaussiana para el estado del objetivo:

$$X_t \sim \mathcal{N}(\hat{X}_t, P_t)$$

donde:

- ▶ \hat{X}_t : Media de la distribución.
- ▶ P_t : Matriz de covarianzas.

Para ello se integran las siguientes fuentes de información:

- ▶ La distribución para X_t dada por el modelo dinámico.
- ▶ La *historia* de medidas (Z_0, \dots, Z_t).

Proceso estocástico subyacente

El filtro de Kalman supone que el proceso que queremos estimar está gobernado por la ecuación:

$$X_t = AX_{t-1} + Du_t + w_{t-1}$$

- ▶ A relaciona el estado X_t con X_{t-1} en ausencia de ruido.
- ▶ D es la matriz que relaciona un vector de control u_t con X_t . Nosotros prescindiremos de estos factores.
- ▶ w_{t-1} es el factor aleatorio que representa el error del proceso estocástico.

Proceso estocástico subyacente

También suponemos que las medidas tomadas son ruidosas:

$$Z_t = HX_t + v_t$$

- ▶ H relaciona la medida Z_t con el estado subyacente del modelo X_t .
- ▶ v_t es el factor aleatorio que representa el error que se produce en el proceso de medida.

Proceso estocástico subyacente

Los vectores aleatorios w_t y v_t deben ser independientes y gaussianos:

$$p(w) \sim \mathcal{N}(0, Q)$$

$$p(v) \sim \mathcal{N}(0, R)$$

- ▶ Q : covarianza del error de predicción.
- ▶ R : covarianza del error de medida.

Notación

- ▶ \hat{X}_t^- : Estimación **a priori**, dada por la dinámica.
- ▶ \hat{X}_t : Estimación **a posteriori**, dada la medida Z_t .
- ▶ Error **a priori**:

$$e_t^- \equiv X_t - \hat{X}_t^-$$

- ▶ Error **a posteriori**:

$$e_t \equiv X_t - \hat{X}_t$$

- ▶ Covarianza del error **a priori**:

$$\mathsf{P}_t^- = \mathcal{E}[e_t^- e_t^{-T}]$$

- ▶ Covarianza del error **a posteriori**:

$$\mathsf{P}_t = \mathcal{E}[e_t e_t^T]$$

Objetivo del algoritmo

Calcular \hat{X}_t como una combinación lineal entre \hat{X}_t^- y una diferencia ponderada entre la medida actual Z_t y una predicción de medida $H\hat{X}_t^-$.

$$\hat{X}_t = \hat{X}_t^- + K(Z_t - H\hat{X}_t^-)$$

- ▶ A la diferencia $(Z_t - H\hat{X}_t^-)$ se la denomina **innovación** o **residuo**.
- ▶ El residuo representa la discrepancia entre la medida predicha y la realmente encontrada.
- ▶ La matriz K (**ganancia de Kalman**), se escoge para que minimice la covarianza del error *a posteriori* P_t .

Cálculo de la ganancia de Kalman

Una forma para K que minimiza el valor de P_t es:

$$K = P_t^- H^T (H P_t^- H^T + R)^{-1}$$

donde R es la covarianza del error de medida.

Observaciones

- ▶ Cuando R tiende a cero, la medida obtenida es muy buena, entonces K aumenta, dando mayor peso al residuo:

$$\lim_{R \rightarrow 0} K = H^{-1}$$

- ▶ Cuando la covarianza del error *a priori* P_t^- tiende a cero, entonces la predicción es muy buena, y K disminuye, dando menos peso al residuo:

$$\lim_{P_t^- \rightarrow 0} K = 0$$

Algoritmo para el filtro de Kalman discreto

1. Predicción:

$$\hat{X}_t^- = A\hat{X}_{t-1}$$

$$P_t^- = AP_{t-1}A^T + Q$$

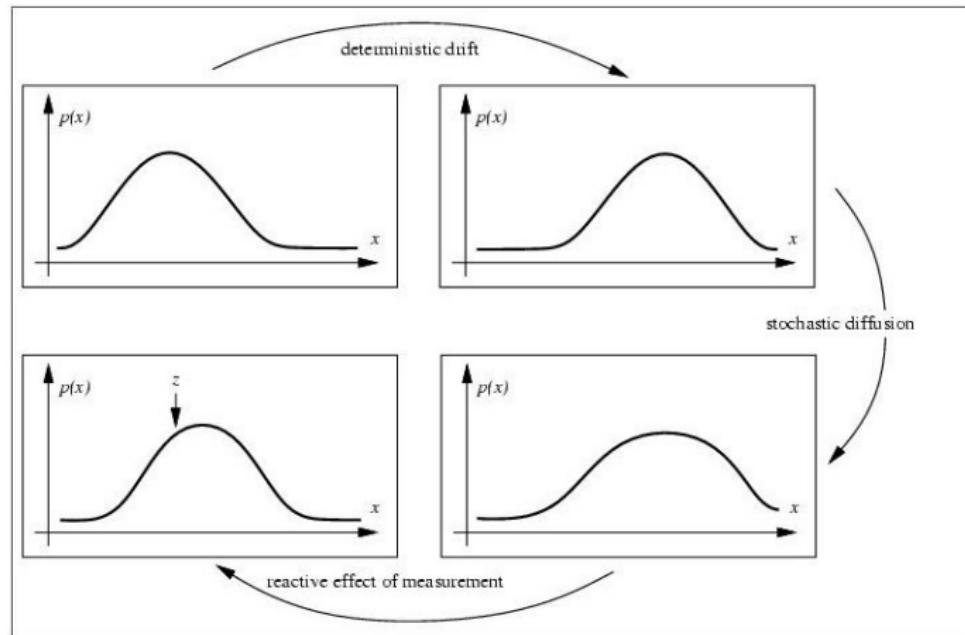
2. Corrección:

$$K = P_t^- H^T (H P_t^- H^T + R)^{-1}$$

$$\hat{X}_t = \hat{X}_t^- + K(Z_t - H\hat{X}_t^-)$$

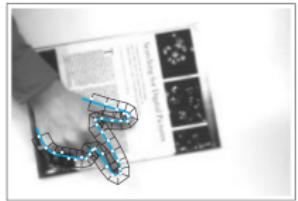
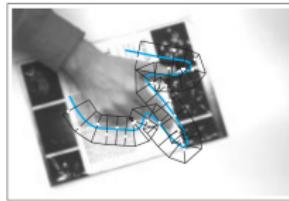
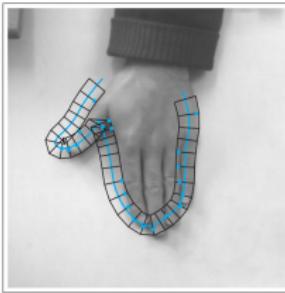
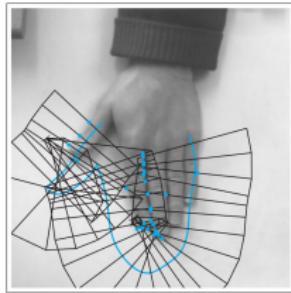
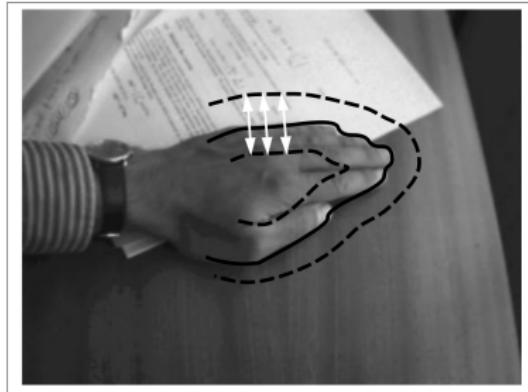
$$P_t = (I - KH)P_t^-$$

El filtro de Kalman como propagación de densidad



Representación gráfica del funcionamiento del Filtro de Kalman.

Ejemplos de seguimiento con filtro de Kalman



Limitaciones del filtro de Kalman

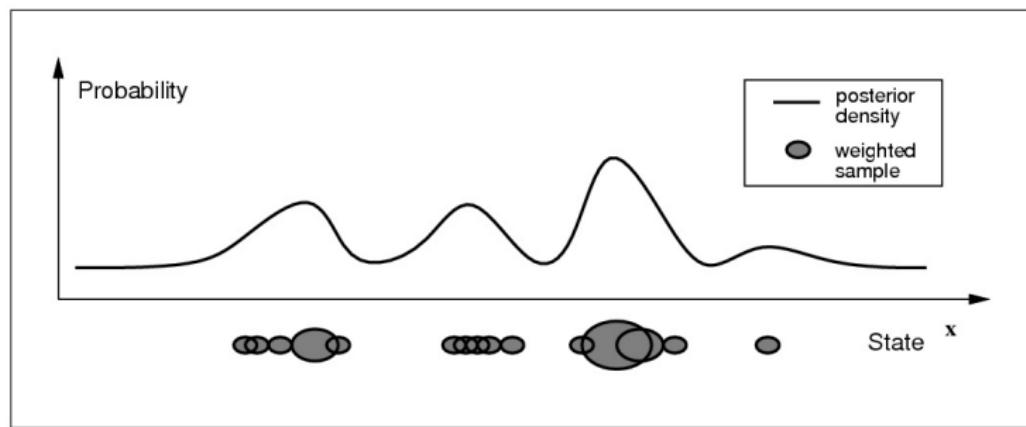
- ▶ Filtro de Kalman \Rightarrow Distribuciones Gaussianas.
- ▶ En situaciones de seguimiento reales, $p(X)$:
 - ▶ No puede aproximarse bien con una Gaussiana.
 - ▶ Es multimodal (cuando hay objetos parecidos).
- ▶ En estos casos, el filtro de Kalman deja de ser óptimo, o de funcionar por completo.

Algoritmos de filtrado de partículas

- ▶ Si la distribución es Gaussiana (Kalman), queda definida por:
 - ▶ La media \hat{X} .
 - ▶ La matriz de covarianzas P .
- ▶ Cualquier distribución puede ser representada mediante una nube de **partículas** (x, π):
 - ▶ x es la posición asociada a la partícula en el espacio de características.
 - ▶ π es el peso de la partícula.
- ▶ Cuanto mayor sea el número de partículas, mejor es la aproximación.
- ▶ Los **algoritmos de filtrado de partículas** usan esta aproximación.

Aproximación de la distribución de densidad

Las muestras correspondientes a valores más altos de la densidad tienen mayor peso:



Muestreo factorizado

- ▶ Usa $p(X)$ para generar una distribución que aproxima la distribución **a posteriori** $p(X|Z)$.

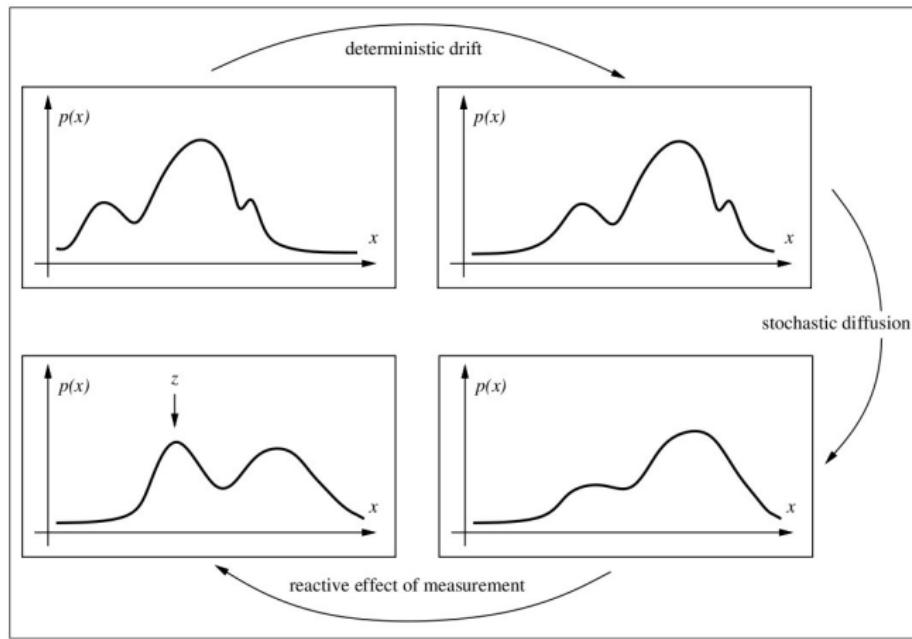
- ▶ Pasos:

1. Se genera un conjunto $\{x_1, \dots, x_n\}$ a partir de $p(X)$.
2. Los pesos π_i se calculan:

$$\pi_i = \frac{p(Z|X = x_i)}{\sum_{j=1}^n p(Z|X = x_j)}$$

- ▶ El conjunto $\{(x_1, \pi_1), (x_2, \pi_2), \dots, (x_n, \pi_n)\}$ aproxima la distribución $p(X|Z)$.

Propagación de densidad con CONDENSATION



Representación gráfica del funcionamiento de CONDENSATION.

Descripción del algoritmo

1. Muestras en $t - 1$:

$$C_{t-1} = \{(x_1^{(t-1)}, \pi_1^{(t-1)}), \dots, (x_n^{(t-1)}, \pi_n^{(t-1)})\}$$

2. Selección:

- ▶ Muestrear n valores x'_i a partir de C_{t-1} .

3. Predicción:

- ▶ Generar x_i a partir de x'_i , empleando la dinámica.

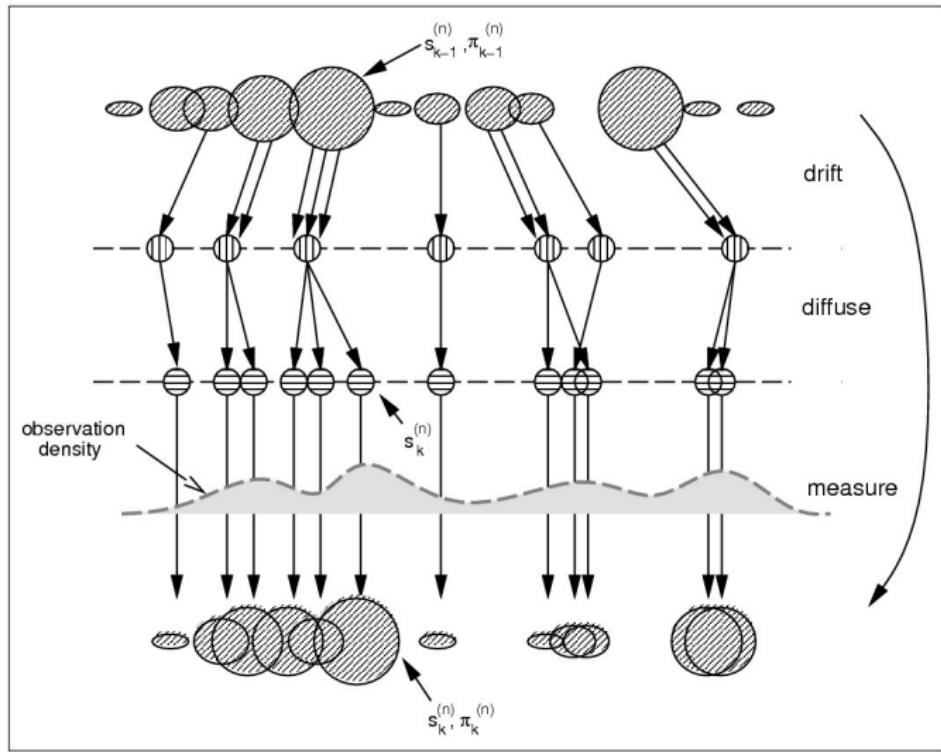
4. Medida:

- ▶ $\pi_i = p(Z|X = x_i)$.

5. Muestras en t :

$$C_t = \{(x_1, \pi_1), \dots, (x_n, \pi_n)\}$$

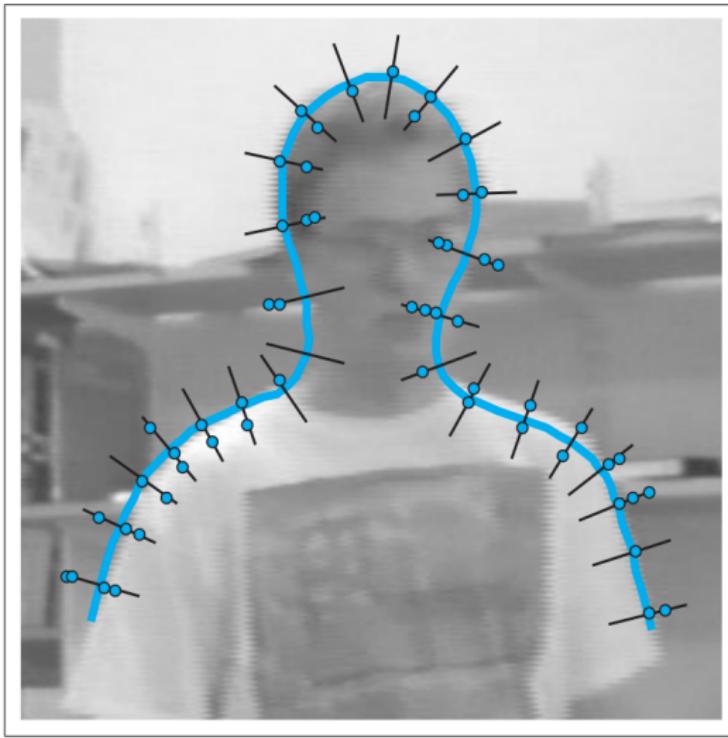
Representación gráfica de CONDENSATION



Modelos de observación

- ▶ Proporcionan la probabilidad de que se dé una observación, dada la posición del objeto, $p(Z|X = x_i)$
- ▶ Basados en fronteras: trazando perpendiculares al contorno y buscando extremos en la derivada de la imagen.
- ▶ Dado que, para emplear dinámicas de segundo orden, se incluyen las posiciones del objeto en $t - 1$ y $t - 2$, se pueden construir modelos basados en movimiento (flujo óptico).
- ▶ Los distintos z deben ser estadísticamente independientes, por lo que han de escogerse lo suficientemente separados como para que se pueda considerar cierta esta condición.

Ejemplo de modelo de observación

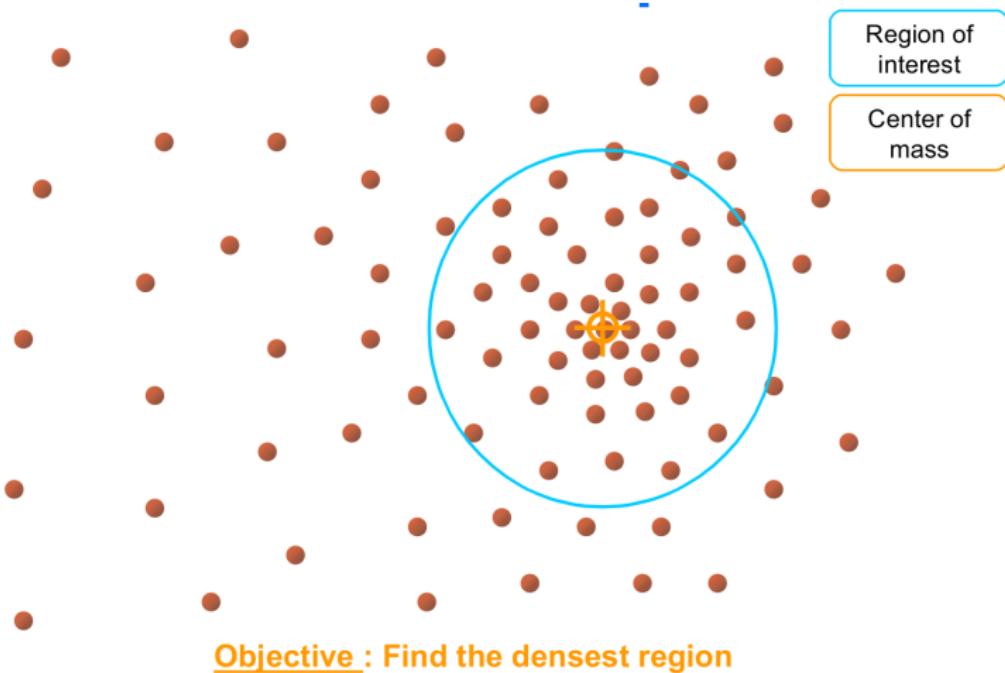


Filtrado de núcleo (*Kernel Tracking*)

- ▶ Eficiente para seguir objetos cuya apariencia viene dada por un histograma.
- ▶ El histograma caracteriza un objeto a partir de una distribución de colores.
- ▶ Es útil para seguir objetos no rígidos, ya que se prescinde de la componente espacial dentro del área que ocupa el objeto.
- ▶ Los algoritmos de seguimiento se basan en el procedimiento conocido como *mean shift*.
- ▶ En principio, el vector **X** que caracteriza al objeto son únicamente sus coordenadas en la imagen.

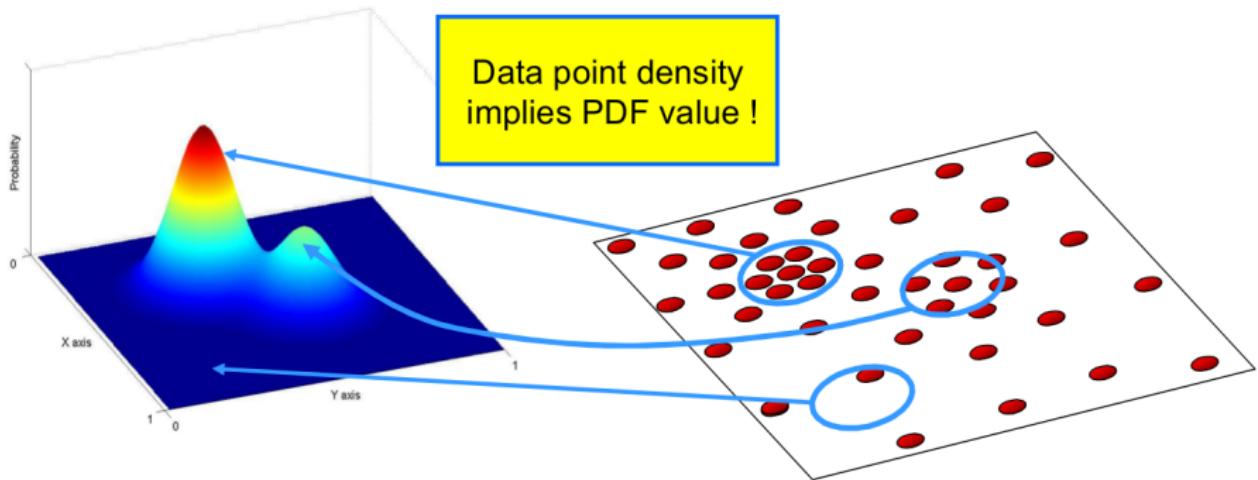
El procedimiento *mean shift*

Descripción intuitiva:



Aproximación de la función de densidad

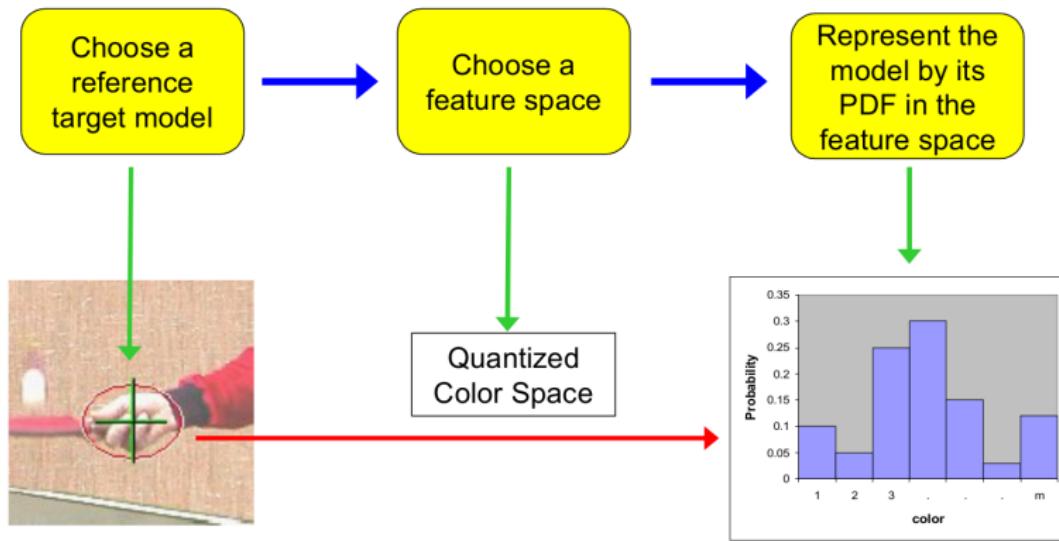
Suponemos que se muestran puntos de una distribución subyacente.



Estimación del máximo

- ▶ Dado el histograma del objeto, podemos calcular la probabilidad de que un píxel pertenezca o no al objeto, que será proporcional al valor de la cubeta correspondiente del histograma.
- ▶ *Mean shift* actúa calculando la media de todas las coordenadas de los píxeles de la región, ponderada por la probabilidad de cada uno.
- ▶ El centro de masas resultante actúa como un gradiente de la distribución de probabilidad subyacente.
- ▶ Una vez alcanzada la convergencia, se puede calcular la *calidad* del ajuste a partir de la distancia entre el histograma de referencia y el de la región obtenida.

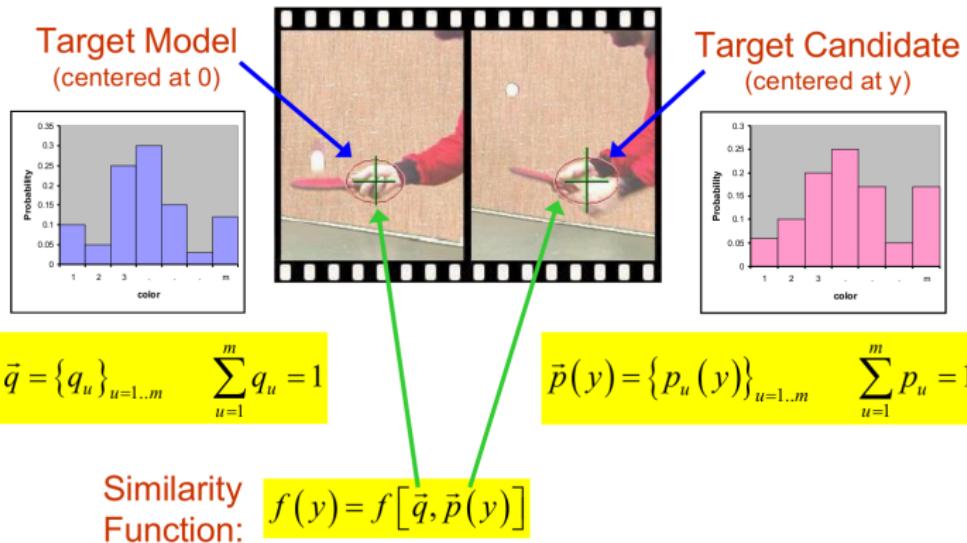
Representación del objeto



Representación del objeto

- ▶ Usualmente se emplea un histograma de color.
- ▶ Se puede hacer *troceando* el cubo RGB. Por ejemplo, con 16 tonos para cada banda se tiene un histograma de $16^3 = 4096$ cubetas.
- ▶ También se emplean los histogramas marginales de cada banda. En este caso, con 16 tonos por banda se obtienen 48 cubetas.
- ▶ A veces se emplean colores *normalizados*, sin la componente de intensidad, para reducir la sensibilidad a la iluminación y las sombras.

Representación de la distribución de probabilidad



- ▶ La función de similaridad $f(\cdot, \cdot)$ nos da la distancia entre el histograma de referencia \vec{q} y el del objeto en la coordenada y , $\vec{p}(y)$.

Ajuste del tamaño de la región que ocupa el objeto

- ▶ El método *mean shift* parte de un área de la imagen de tamaño conocido.
- ▶ Si el objeto se acerca o aleja, el área que ocupa crece o decrece.
- ▶ El algoritmo *camshift*, una vez alcanzada la convergencia, calcula el ajuste para la región original, una un poco más grande y otra un poco más pequeña, todas centradas en el mismo punto, y se queda con la que tenga una menor distancia entre su histograma y el de referencia.
- ▶ Este proceso también se puede aplicar a otros parámetros, como la orientación de una región elíptica o rectangular.