PEDESTRIAN TRACKING FROM A MOVING CAMERA CAR



Alberto del Amo Panadero

Santiago Domínguez Collado

ÍNDICE

INTRODUCCIÓN …………………………………………………………………………………………………………………………3

DESARROLLO ……………………………………………………………………………………………………………………………..4

DETECTOR ……………………………………………………………………………………………………………………..4

FILTRO DE KALMAN ………………………………………………………………………………………………………..5

CONCLUSIONES ..………………………………………………………………………………………………………………………..6

BIBLIOGRAFÍA …………………………………………………………………………………………………………………………….6

3. INTRODUCCIÓN

Durante este trabajo hemos estudiado y modificado el código proporcionado por Matlab de detección de peatones desde una cámara en movimiento. La documentación del propio código, así como la de Matlab han sido nuestras principales fuentes de consulta durante el trabajo.

Partiendo de dicho ejemplo, nuestros objetivos principales fueron los siguientes:

* Comprender el código estudiando la documentación y analizando los múltiples comentarios que venían en el propio archivo, fueron de gran utilidad.
* Modificar el código para que recibiera como entrada el vídeo en tiempo real de la cámara del ordenador en lugar de un vídeo ya grabado.
* Modificar diversos datos para ajustar los parámetros a la nueva entrada de vídeo, buscando la máxima eficiencia del código.
* Analizar y entender los puntos flacos del software para explicarlos durante la exposición.

4. DESARROLLO

En la etapa de desarrollo hacemos dos claras distinciones, el detector de personas y el filtro de Kalman.

4.1 DETECTOR

El detector usado en este software es un detector de personas mediante Agregate Channel Features que ha sido entrenado con 10 horas de video grabadas desde una cámara en un coche.

El detector funciona de manera que, tras recibir una imagen, computa múltiples imágenes a partir de la recibida, y a cada una le realiza una transformación en concreto. Tras este proceso el detector obtiene diversos canales de características gracias a las imágenes computadas, esas son las Channel Features.

De todas estas características, el detector hace un compendio y genera un vector de características que define toda la imagen original.

Posteriormente con ese vector, se usa un algoritmo de clasificación (AdaBoost) para detectar lo que sea demandado, en este caso, peatones.

4.1 FILTRO DE KALMAN

El filtro de Kalman es un algoritmo utilizado identificar el estado de un sistema dinámico lineal sobre el cual no podemos realizar mediciones exactas.

Se compone de dos etapas una de predicción y otra de corrección.

•Φ: Matriz de Transición de estados

•x: El estimado a priori del vector de estados.

•P: Covarianza del error asociada a la estimación a priori.

•z: Vector de mediciones al momento *k*.

•H: La matriz que indica la relación entre mediciones y el estado.

•R:  La matriz de covarianza del ruido de las mediciones.

Predicción:

1. Calculamos el estado en el cual se debería encontrar el sistema sabiendo cual era el estado anterior. Para ello utilizamos la matriz de transición de estados (φ).
2. Calculamos también P que es la varianza del error en el instante de la predicción.

Corrección:

1. Calculamos K (ganancia de kalman) en base al error de las mediciones y el error de la predicción. Utilizaremos K para saber en qué proporción debemos utilizar la predicción o la medida en este instante.
2. Actualizamos la medición yk.
3. Calculamos el estado xk|k a partir de K y la medida obtenida en y la nueva medidayk.
4. Calculamos Pk|k a partir de la matriz H que relaciona las mediciones con los estados, la ganancia y la calculada en el instante anterior.

En nuestro caso el filtro tomaría como mediciones los centroides de los tracks asignados a partir del detector y como modelo de movimiento un espacio en el que los objetos varían su posición con una velocidad lineal.

Al añadir el filtro a nuestro detector conseguimos aumentar la precisión con la que se siguen los peatones y nos permite continuar con él mismo a pesar de que durante unos frames no obtengamos ninguna medición de la posición del objeto.

El filtro también tiene sus desventajas y es que nos puede llevar a perder a los tracks o que estos sean equivocados en caso de que no se muevan con una velocidad lineal. Esto se puede solventar implementando un filtro de Kalman extendido que modele el espacio permitiendo variaciones de posición no lineales.

1. Conclusiones

Como hemos podido, comprobar a pesar de no haber calibrado todos los parámetros del detector y el filtro a nuestro caso de uso, obtenemos una precision buena.

Consideramos que el algoritmo es bastante preciso y se debe tener en cuenta en caso de realizar trabajos de este tipo.

1. Bibliografía

* <https://es.mathworks.com/help/vision/ref/detectpeopleacf.html>
* <https://vision.cornell.edu/se3/wp-content/uploads/2014/09/DollarPAMI14pyramids_0.pdf>
* <http://www.tsc.uc3m.es/~mlazaro/Docencia/Doctorado/FiltAdapt/Kalman.pdf>