Person tracking

Using Kalman filter

Autor: Íñigo Alonso

Introducción

El filtro de Kalman es un algoritmo que permite identificar un estado oculto a través de un estado anterior, unas medidas u obervaciones y unas entradas y el sistema bien modelado. Un ejemplo puede ser la localización através de GPS, donde el sistema modelado es la función que define cómo se está moviendo el usuario, las medidas son las medidas dadas por el GPS y las entradas, podrían ser (si es que hay) fuerzas conocidas que actuen sobre el usuario.

En esta práctica se ha utilizado para mejorar un tracker de personas, que, sin el filtro de Kalman, es tan solo un descriptor HOG, pero gracias a este se puede obtener predicciones incluso cuando no se detecte la persona, mejorar y suavizado del tracking y descartar medidas espúrias.

El modelo del sistema es lineal, lo que simplifica su resolución.

Sistema modelado

El sistema viene dado por estas ecuaciones:

$$
\begin{aligned}
u_{t+1} &= u_t + \dot{u}_t \Delta t + \epsilon_{\dot{u}} \Delta t \\
v_{t+1} &= v_t + \dot{v}_t \Delta t + \epsilon_{\dot{v}} \Delta t \\
\dot{u}_{t+1} &= \dot{u}_t + \epsilon_{\dot{u}} \\
\dot{v}_{t+1} &= \dot{v}_t + \epsilon_{\dot{v}}
\end{aligned}
$$

Donde se modela:

1. La posición X sobre la imagen (píxeles)
2. La posición Y sobre la imagen (píxeles)
3. La velocidad en X sobre la imagen (pix/s
4. La velocidad en X sobre la imagen (pix/s)

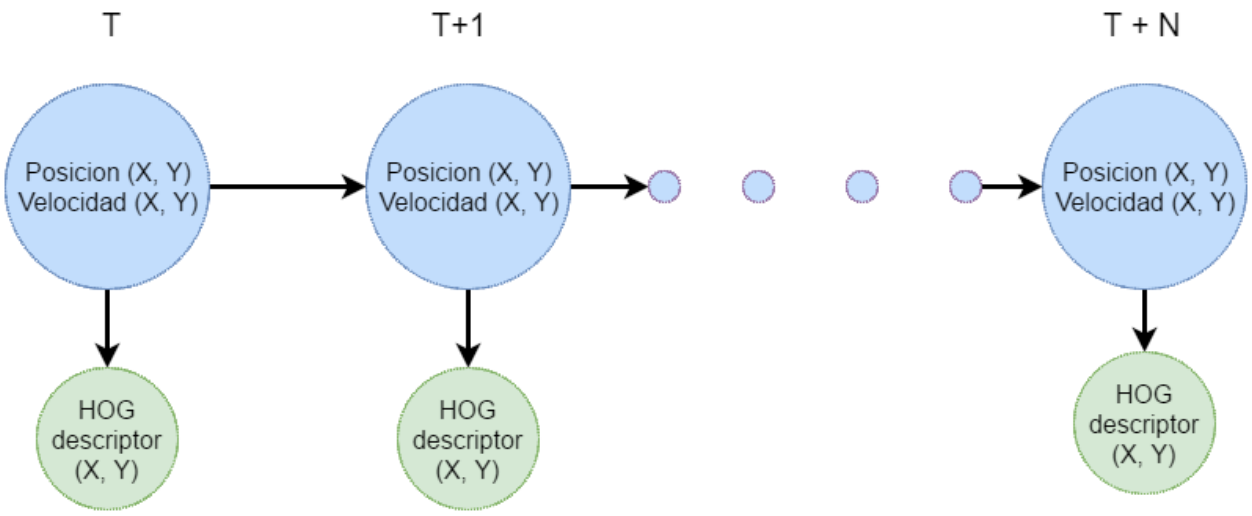
Donde el ruido tiene una distribución normal con media cero y Δt es el tiempo entre frames (1/15 segundos en esta implementación).

Las observaciones se modelan de esta forma:

$$y_t^u = u_t + \delta_u$$
$$y_t^v = v_t + \delta_v$$

Donde el ruido también tiene una distribución normal con media cero y se obtienen a través de una implementación concreta del descriptor de HOG.

Así pues, en este caso, no se tienen entradas al sistema y nuestro modelo del filtro de Kalman se podría representar gráficamente así:



Fijándose en las matrices que definen al modelo en el filtro de Kalman:

Predicción del estado = **A** * VariablesEstado + **B** * entradas + ruido

Predicción de la MatrizCoovarianza = **A** * MatrizCoovarianza * **A'** + **Q**

Medida = **C** * variablesMedida + D * entradas + ruido (En este caso D=0)

También es necesaria la matriz **R** (incertudumbre de la medida)

El sistema tiene estas matrices:

Q

X_t	Y_t	Vx_t	Vy_t	
$(Evx* \Delta t)^2$	0	Δt	0	X_{t+1}
0	$(Evy* \Delta t)^2$	0	Δt	Y_{t+1}
Δt	0	$(Evx)^2$	0	Vx_{t+1}
0	Δt	0	$(Evy)^2$	Vy_{t+1}

Donde Q es la matriz que define la incertidumbre de nuestro modelado del sistema. Evx es la incertidumbre error en la variación de velocidad del sistema en el eje X y Evy en el eje Y.

Esto se deduce de la ecuación del sistema. Los valores concretos en esta implementación son:

- Δt=1/15 segundos, debido a que hay 15 frames por cada segundo real.

- Evx=50 pixeles/segundo. Se ha llegado a este valor primero por una aproximación viendo el tamaño de las imágenes y viendo que en 2 segundos un individuo andando puede cruzar la acera (500 pixeles). Un caso pero sería que a esa velocidad de 250pixeles/segundo en el eje de la X cambiase hacia el otro sentido, pero, pensando en una aproximación sin ser el caso peor, sería que está andando y de repente se para, pero tampoco de una forma muy brusca, lo que por ejemplo podría ser una incertidumbre de cambiar, de un segundo a otro de 150 pixeles/segundo a 0 pixeles/segundo, pero mostrando la información útil durante la ejecución de pruebas, se ha visto que para estos casos un valor de 50 pixeles/segundo es suficiente, ya que en estos escenarios, la variación tampoco va a ser tan alta porque son casos bastante predecibles (un usuario cuando está cruzando, no va a cambiar de opinión e irse atrás) aunque, habría que probar con muchos más datos de prueba viendo que la gente que cruza corriendo, se detecta bien con este valor

- Evy=7 pixeles/segundo. Misma explicación que el caso anterior, pero referente al eje Y.

A

X_t	Y_t	Vx_t	Vy_t	
1	0	Δt	0	X_{t+1}
0	1	0	Δt	Y_{t+1}
0	0	1	0	Vx_{t+1}
0	0	0	1	Vy_{t+1}

Esto se deduce de la ecuación del sistema. Para poner algo más claro porqué se deduce, o cómo, o el significado de la matriz, esta sería la distribución en la que están las varaibles:

X_t	Y_t	Vx_t	Vy_t	
1	0	Δt	0	X_{t+1}
0	1	0	Δt	Y_{t+1}
0	0	1	0	Vx_{t+1}
0	0	0	1	Vy_{t+1}

C

1	0	0	0
0	1	0	0

Deducible de la ecuación de las medidas

R

Ex^2	0
0	Ey^2

Donde Ex es el error que puede llegar a tener el detector de Hog en la persona sobre el eje X y Ey sobre el eje Y. Los valores concretos en esta implementación son:

- Ex= Ey=8 píxeles. A pesar, que el rectángulo donde se muestra la detección es demasiado grande, el centro de este, el cual representaría el centro de la persona, es bastante preciso, llegándose a equivocar muy poco, por eso se le ha dado un valor tan bajo

Obtención del estado inicial

El estado inicial del sistema se obtiene cuando llega una primera imagen con una persona detectada.

La posición inicial, se toma como la que se obtiene por la medida del descriptor de HOG, y la velocidad se pone a cero, ya que, el propio sistema la irá aprendiendo a través del modelo del sistema.

La matriz de covarianzas, se toman valores que concuerden con lo que hemos descrito en el estado inicial, es decir, los valores de covarianza de la posición, se parecerán a los descritos por la matriz de covarianzas del detector, ya que se coge de ahí la posición, y como hemos puesto una velocidad de cero, se pone una covarianza alta.

Estos últimos valores tampoco son muy importantes, puesto que lo que importa es modelar bien las del sistema, ya que es ahí donde irá convergiendo.

Consideraciones

¿Cuándo se decide parar de trackear a la persona?

Se podrían aplicar varias técnicas, pero se ha decido juntar dos ideas básicas. La primera idea es dejar de seguir a la persona cuando llevas un número de frames sin verla, por ejemplo 25 frames. La segunda es, cuando la desviación típica en la matriz de covarianzas de la posición X e Y supera un valor concreto, en este caso, 250 píxeles, también se supone que se para de trackearla porque se la considera perdida.

¿Cómo se hace el matching?

Lo primero de todo es que exista una persona en la imagen, a continuación, se calcula S (residual covariance) y se compara, para una distribución normal, en el 0.99, si el dato es válido pensando en la dimensión de la posición X y de la posición Y.

Deducciones

En este sistema, ¿Qué se puede concluir respecto a los valores posibles que se pueden dar a las matrices de covarianza?

Lo primero de todo, es que cuanta menos incertidumbre exista, mejor. Cuanta más incertidumbre haya en la medida, el update, te servirá de menos porque apenas afectará a tu predicción, y cuanta más incertidumbre tenga tu modelo, menos te fiarás de tu predicción.

Respecto a esta práctica del person tracker, se tiene un detector de HOG que define mejor que el modelo donde estás, dado que se equivoca muy pocos píxeles, pero, la incertidumbre de lo que puede hacer una persona (cambios de velocidad, de dirección.) es menos fiable.

Cuanta más incertidumbre se ponga en la matriz de covarianzas del sistema, mayor incertidumbre tendrás cuando no haya medidas y no tengas el update y serás más flexible, pero, por el contrario, tener un valor demasiado alto, hará que no tengas en cuenta tanto tu modelo y más la medida, y estarás perdiendo información si lo modelas mal.

Lo mismo pasa con el detector de HOG, si se da un valor más alto de lo debido, estarás perdiendo información ya que no afectará tanto en la fase de update.

%voy a plantearlo por falta de tiempo

Opcional

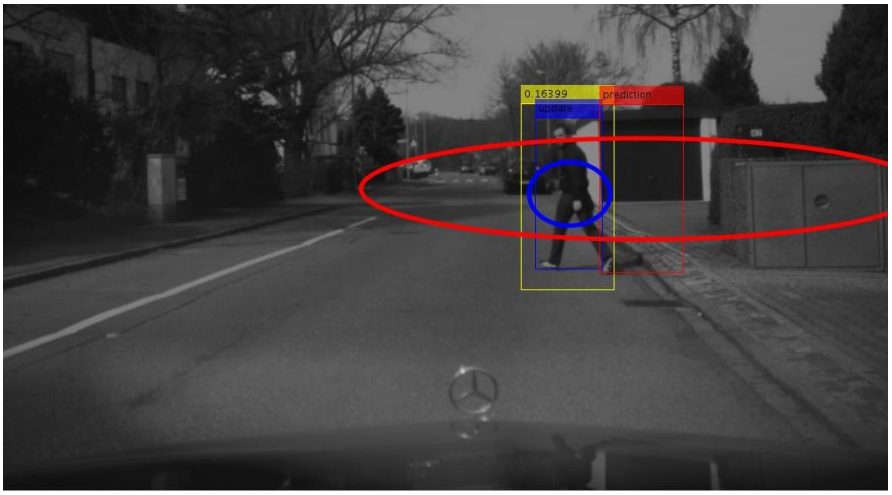
El apartado opcional no se ha implementado por falta de tiempo y solo se han planteado algunas anotaciones que serían de base para realizarlo.

Conceptualmente no va más allá de tener en cuenta ciertos aspectos y consideraciones extra.

- Cada persona trackeada estaría guardada en una estructura de datos con sus estados y variables correspondientes.
- En el matching habría que tener varias consideraciones. Lo primero, habría que hacer un matching de las personas que estabas trackeando, empezando a hacer el matching entre las detecciones y las predicciones cuya distancia sea menor entre ellas, y siempre teniendo en cuenta cuando descartar matchings (apartado consideraciones, ¿Cómo se hace el matching?). Después, las predicciones que no tengan pareja (y el matching del apartado

consideraciones tampoco se cumple), se dan como personas que ya no hay que trackear más, y las detecciones que no tienen pareja, se considerarían como personas nuevas a trackear.

Resultados



En esta foto se puede apreciar justo el momento en el que tras varios frames sin detectar a la persona, se vuelve a detectar.

La elipse roja representa la incertidumbre de la predicción, ya que lelva un tiempo sin ver a la persona, y la azul, la del update. Los rectángulos, representan dónde está la persona, también rojo predicción y azul update respectivamente. El rectángulo amarillo es la medida del descriptor de HOG.

En esta, se ve un filtrado en un frame con la predicción y el update.

Aquí se puede apreciar el recorrido realizado por la predicción y por el update. Se puede apreciar que el update suaviza las lo que se obtiene de las medidas y un poco la predicción, y en el caso en el que el detector diera medidas falsas o erróneas o no fuera muy fiable, se apreciaría un mayor suavizado entre la predicción y la medida. También se puede apreciar que justo en el momento que ser pierde la pista al peatón es cuadno gira.

Esta es una imagen en la que tan solo se ha podido hacer la predicción y no se tiene ni medida ni por lo tanto un update.