Universidad Zaragoza
1542

Shamuel Manuel Manrrique Aquino
NIP: 802400

Kalman Filter

Resumen

Esta práctica consiste en el desarrollo de un modelo escrito en python para el seguimiento de personas en una secuencia de imágenes usando filtros de kalman también conocidos como Estimación Lineal Cuadrática (LQE por sus siglas en inglés). El objetivo es encontrar un punto medio entre la estimación aportada por un agente externo estático que puede contener ruido y ser imprecisos y el modelo. Este estimador permite actualizar y mejorar la predicción con un rango de error cada vez menor.

Objetivo

El objetivo de la práctica es implementar un modelo que siga el desplazamiento de personas en una secuencia de imágenes mostrando la trayectoria de la misma sin importar si esta se aleja o se acerca a la cámara, su busca una predicción que contenga menor rango de error.

Implementación

Para el desarrollo de este práctica se usó el lenguaje de programación python version 3.7, y librerías como opencv, numpy, imutils, Júpiter Notebook, etc. La elaboración de la práctica se puede seccionar en las siguientes etapas descritas a continuación:

1. Pasos previos a la implementación del algoritmo.Primero se obtienen las rutas de las imágenes con las que se trabajará usando la librería glob que permite el uso de wildcards, se guarda las imágenes en un arreglo. Creamos un for para leer el arreglo de imágenes, modificamos el tamaño para mejorar la detección, mientras más pequeña la imagen(width 300) se puede observar un número menor de detenciones y menor alcance de la detección por la cantidad de pixel. En el caso opuesto se muestra un mayor número de detecciones y también mayor alcance. Se determinó un tamaño de 1200.
2. Ajustes y implementacion del detector. Se crea una copia de la imagen y se cambia el color de la misma a gris para crear más contraste y mejorar la detección usando la función HOG (Histogramas Orientados a Gradientes) de la librería cv2 que permite que le sea indicado el objeto que se quiere detectar en este caso una persona.

Universidad Zaragoza
1542

Shamuel Manuel Manrrique Aquino
NIP: 802400

3. Una vez inicializado el HOG al pasarle la imagen se busca a partir de una distribución de gradientes de intensidad la detección, es decir, por las direcciones de los bordes. Básicamente esta caracterización se puede lograr dividiendo la imagen en regiones pequeñas y conectadas, luego calcula los gradientes internos. Terminado el proceso la función detectMultiScale devuelve dos arreglos el primero contiene cuatro puntos que son la posición en x e y, además de weight, height. El segundo arreglo contienen el peso asociado al rectángulo anterior. Este detector puede contener errores y un nivel de confianza que varía de un ambiente a otro para reducir esto se aplica Filtro Kalman para corregir las mediciones erróneas y predecir la ubicación en cuadros donde no hay mediciones disponibles.

4. Implementación del algoritmo del filtro de kalman que consta de dos pasos: el primero la predicción en el cual el sistema predice dada una identificación previa y el segundo paso es el de actualización donde el sistema realiza una estimación dada una periodo de tiempo.

Se plantea un sistema para la detección de personas en imagen de forma continua al cual queremos aplicar Filtro de Kalman para intentar estimar el estado de la persona en un tiempo determinado. Dado que no conocemos cómo se distribuyen los datos se procede a usar la Distribución Gaussiana dado que este tipo de distribución nos permite trabajar con trocitos lineales (usando las ecuaciones a continuación) a pesar de que el problema no es lineal

- $\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{a}_t + \epsilon; \qquad \epsilon \sim \mathcal{N}(0, Q)$
- $\boldsymbol{y}_t = C\boldsymbol{x}_t + D\boldsymbol{a}_t + \delta; \qquad \delta \sim \mathcal{N}(0, R)$

Referencia: Tomadas de las láminas de clase del curso Sistemas Inteligente unizar.

La variables epsilon y delta representan el error estimado y la medida del ruido del sensor respectivamente. Estas variables son independientes y se distribuyen como epsilon(media 0, varianza Q) y delta(media 0, varianza R) .

Universidad Zaragoza
1542

Shamuel Manuel Manrrique Aquino
NIP: 802400

Para proceder a aplicar el algoritmo de KF procedo a definir el conjunto de matrices derivadas del enunciado de la práctica.

```
# init sigma
S0 = np.zeros(shape=(4, 4), dtype=np.float32)

# State to the measurement
C = np.diag([1.0, 1.0, 1.0, 1.0])

# Transition matrix
A = np.array([[1, 0, dt, 0],
              [0, 1, 0, dt],
              [0, 0, 1, 0],
              [0, 0, 0, 1]], np.float32)

# Input effect matrix
B = np.array([[dt**2, 0],
              [0, dt**2],
              [0, 0],
              [0, 0]], np.float32)

# Constant matrix to multiply B
a = np.array([[4], [3]], np.float32)


# Noices of covariances matrix
Q = np.array([[dq, 0, 0, 0],
              [0, dq, 0, 0],
              [0, 0, dq, 0],
              [0, 0, 0, dq]], np.float32)

# Mesurement error matrix
R = np.array([[dr, 0, 0, 0],
              [0, dr, 0, 0],
              [0, 0, dr, 0],
              [0, 0, 0, dr]], np.float32)
```

Los valores de los delta en la matrices A, Q, R se consideran constantes, pese a esto es importante resaltar que el comportamiento del modelo varía según el valor asignado el **“dt”** = 1/fps que es la cantidad de **“pfs”**=24, valor de **“dq”** es que tan rápido crece la incertidumbre del modelo cuando no recibe detección, mientras más grande el valor más rápido el círculo alrededor de la persona los valores de dq=90-110 se comportan bien. El **“dr”** qué tan confiable es la detección del sistema mientras más pequeño el valor se verá un simple punto dada la confianza depositada en el detector, si se incrementa el radio del círculo también lo hará.

Universidad Zaragoza
1542

Shamuel Manuel Manrrique Aquino
NIP: 802400

Para el paso de predicción usamos la siguiente ecuaciones derivadas de las mostradas anteriormente:

- $\mu_{t|t-1} = A\mu_{t-1} + \mathrm{B}\boldsymbol{a}_t$
- $\Sigma_{t|t-1} = A\Sigma_{t-1}A^T + Q$

Para el paso de Actualización:

- $\mu_{t|t} = \mu_{t|t-1} + K_t r_t$
- $\Sigma_{t|t} = (I - K_t C)\Sigma_{t|t-1}$

r_t is the residual $r_t \triangleq \boldsymbol{y}_t - \hat{y}_t$, being $\hat{y}_t \triangleq C\mu_{t|t-1} + \mathrm{D}\boldsymbol{a}_t$

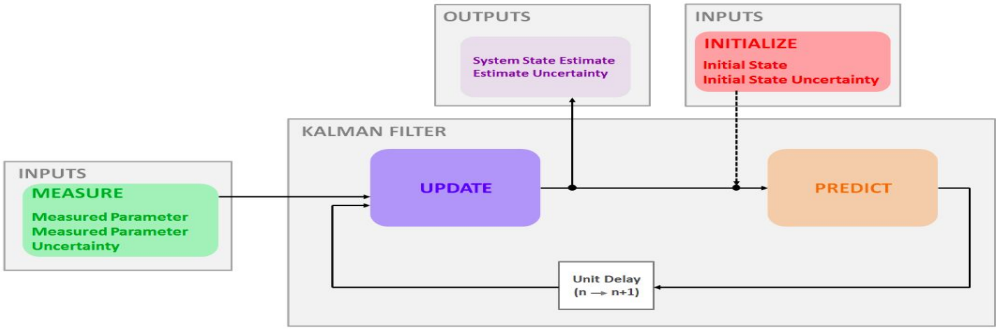
K_t is called the Kalman gain $K_t \triangleq \Sigma_{t|t-1}C^T S^{-1}$, and S the residual covariance $S_t = C\Sigma_{t|t-1}C^T + R$

Fuente: Fórmula extraída de las láminas de clase de Sistema inteligente

donde
A es la matriz de transición.
B es la matriz de efecto de entrada.
Ut(miu) y **St**(sigma) son para predecir la media y la covarianza iterativamente..
y es la medición tomada en el paso t.
rt es la medida residual en el tiempo t.
s la medida de medición de covarianza en el tiempo del paso t.
k Filtro Kalman, indica el modelo la corrección que debería aplicar en el paso t.

Funcionamiento del kalman Filter

Para la explicación del funcionamiento del modelo se tomó la siguiente imagen extraída de la página de Kalman FIlter in one direction con referencia en el pie de página.



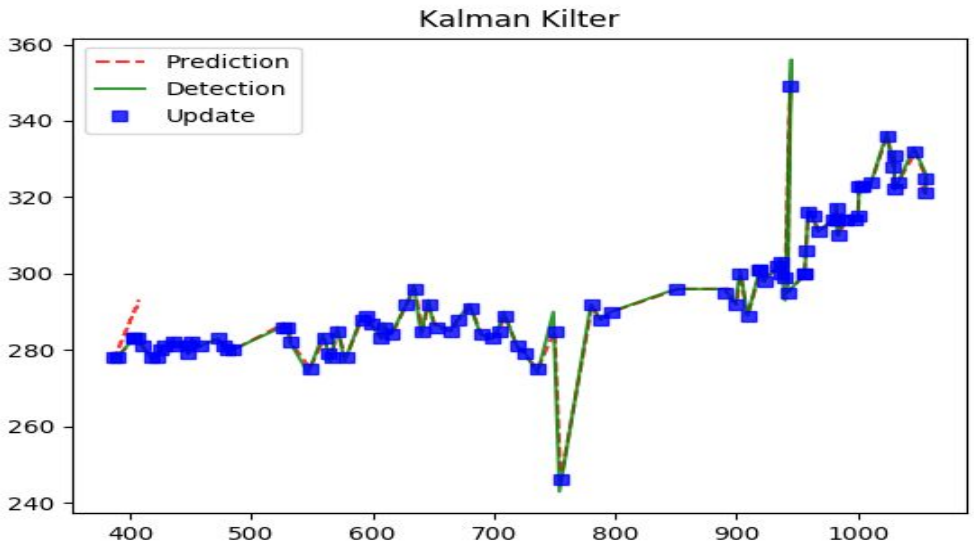
Referencia: https://www.kalmanfilter.net/kalman1d.html

Shamuel Manuel Manrrique Aquino
NIP: 802400

El sistema inicia y el modelo va a arrancar su trabajo después de la primera detección donde se obtiene los valores de la posición de la persona en el eje x e y. Obtenemos el primer U0 los valores de la velocidad se asignan en 0 y tomamos la sigma inicial como una matriz de cero asumiendo que como en nuestro primer estado detectado no tiene error. Una vez obtenido estos valores iniciales procedemos a aplicar el Filtro de Kalman iterativamente. Si el detector no detecta a la persona nuestro modelo se fija en la última detención incrementando el radio focales de la elipse (crecimiento de la incertidumbre). Cuando el sistema capta otra vez a una persona la incerteza disminuye. Esto ocurre dado que el modelo está aprendiendo y siempre toma el update como referencia que le indica hacia donde debería tender. Adjunto la gráfica que muestra el comportamiento del modelo.

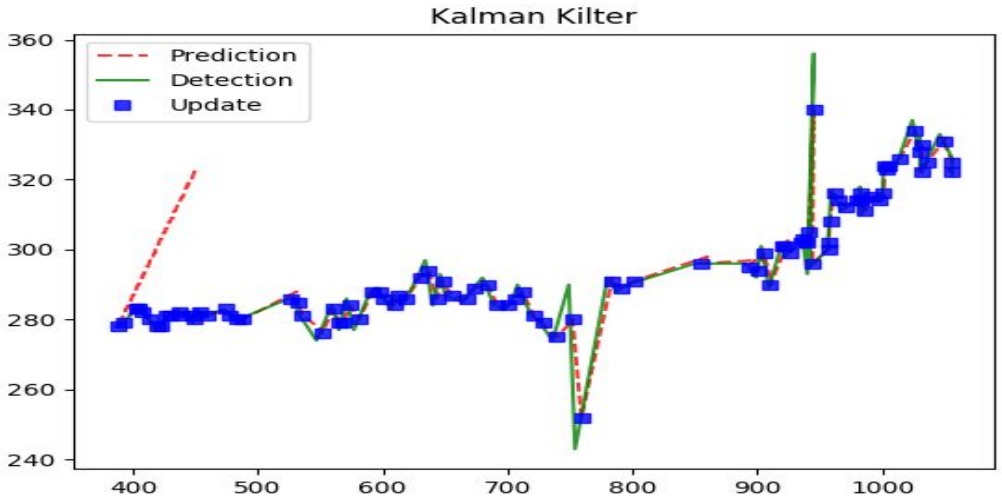
Terminado el modelo se hicieron múltiples pruebas variando los **dt,dq,dr** para analizar el comportamiento del mismo con distintas variaciones en estos valores. Las imágenes se encuentran identificadas con el valor de **dt, dq, dr** con el que fueron generadas(existen otras pruebas en la carpeta plot del proyecto). Con base en los resultados obtenidos en las gráfica y los cambios en la corrida se puede concluir que al incrementar **dr** constante que representa los grados de confianza del detector su tamaño condiciona el tamaño del círculo a mostrar en la detección, da buenos resultados con un valor entre 10-15. Por otro lado el **dq** arroja buenos resultados con valores entre 75-95 si asignas fuera ese rango se puede observar en las gráficas que el error del modelo crece o si es muy pequeño la incertesa crece muy lentamente y la persona queda fuera del rango por un tiempo determinado, lo que representa un error en el modelo que se puede evitar. Por último el **dt** arrojó mejores resultados con un rango entre 1/30-40 fps el tamaño del dt determina la magnitud de la línea roja al final de la predicción. El mejor resultado del modelo se obtuvo con los valores de la figura 1, en resto la predicción tiene mayor rango de error.

Plot Kalman FIlter con dr=11, dq=80, dt=0.025



Shamuel Manuel Manrrique Aquino
NIP: 802400

Plot Kalman FIlter con dr=30, dq=20, dt=0.0416



Plot Kalman FIlter con una matriz 2x1 a =[[3],[4]], dr=11, dq=80, dt=0.02