Capítulo 3. TÉCNICAS DE SUSTRACCIÓN DE BACKGROUND

La técnica de sustracción de background ("background subtraction") es una técnica bastante utilizada para detectar objetos como la diferencia entre un pixel actual y otro de referencia, llamado modelo de background o imagen de background (imagen de fondo). Las áreas donde esta diferencia es significativa indica la localización de un objeto:

$$|Background_t - Frame_t| > \tau \quad (6)$$

donde $\boldsymbol{\tau}$ es un umbral predefinido.

Este proyecto se centrará en los algoritmos que presuponen que el background no varía demasiado con el tiempo (background estático) y que puede ser capturado a priori. Los métodos más representativos para separar el background de los posibles objetos, según Piccardi, M en [3], son:

- Métodos básicos
- Running Average
- Running Gaussian Average
- Mezcla de Gaussianas (Mixture of Gaussians)
- Estimación de densidad del Kernel ("Kernel Density Estimation")
- Estimación basada en la técnica mean-shift
- Aproximación secuencial de densidad del Kernel ("Sequential Kernel Density Approximation)
- Coocurrencias de variaciones en la imagen
- Autobackgrounds ("Eigenbackgrounds")

3.1. Métodos básicos

3.1.1. Diferencia entre píxeles

El background se actualiza simplemente como la nube de puntos del instante anterior al actual:

$$B_t = F_{t-1} \tag{7}$$

donde F_t y B_t son los valores de los pixeles en el instante t y la imagen de background, respectivamente.

Cada cierto tiempo t, cada valor de los pixeles F_t se clasifica como pixel de foreground si cumple la desigualdad:

$$|B_t - F_t| > \tau \tag{8}$$

En otro caso, F_t se clasifica como pixel de background.

El procedimiento de este método se muestra en la *Figura 7*:

- Inicialmente, se toma como background la nube de puntos primera. Seguidamente, se detecta la siguiente nube de puntos y se comprueba si se verifica la desigualdad (8) para determinar si el pixel es foreground (objeto) o background.
- Si un pixel F_t se clasifica como foreground entonces se ignora en el modelo de background, es decir, $B_t = B_{t-1}$. Este pixel F_t se almacena como pixel objeto ($O_t = F_t$).
- Si un pixel F_t se clasifica como background, entonces el background se actualiza como en (7).

Como puede observarse, este método es muy sensible al umbral τ.

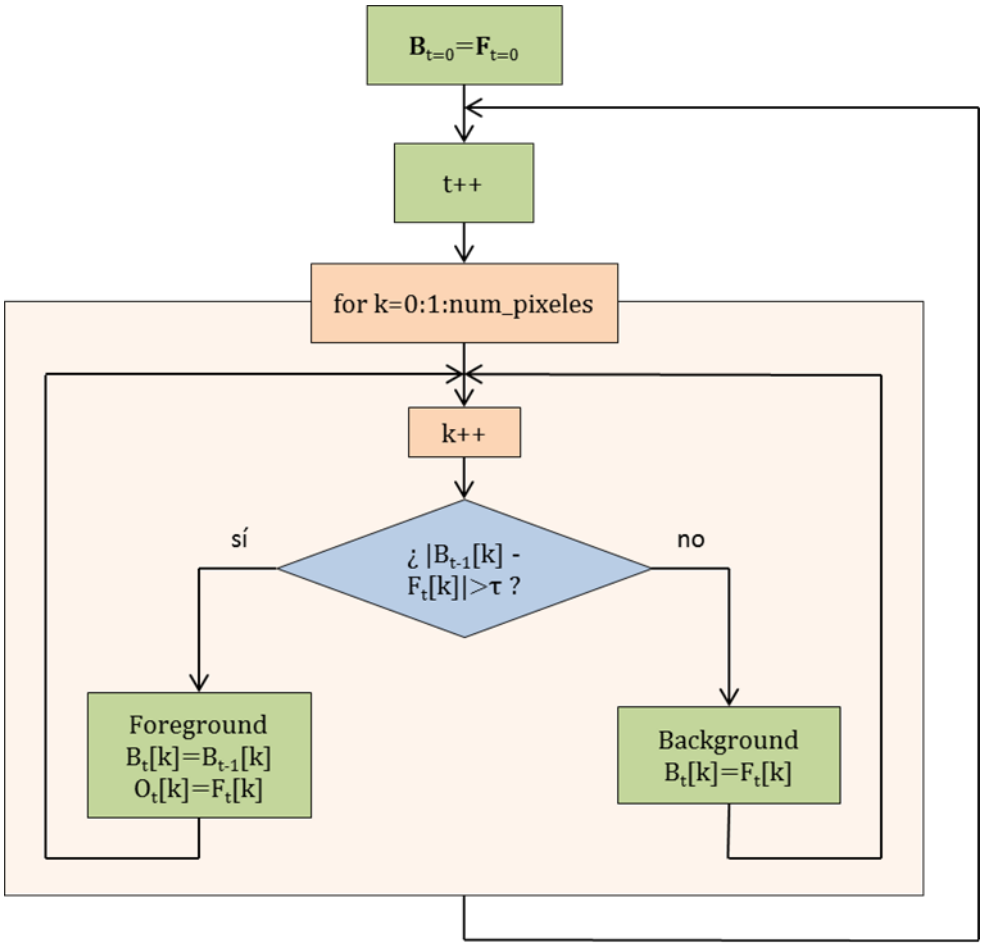

Figura 7: Diagrama del método diferencia entre pixeles

3.1.2. Media de n píxeles

El background se actualiza como la media de los pixeles en n instantes seguidos:

$$B_t = media(F_t, F_{t+1}, \ldots, F_{t+n}) \tag{9}$$

donde F_t y B_t son los valores de los pixeles en el instante t y la imagen de background, respectivamente.

Cada cierto tiempo t, cada valor de los pixeles F_t se clasifica como pixel foreground si cumple la desigualdad:

$$|B_t - F_t| > \tau \tag{10}$$

En otro caso, F_t se clasifica como pixel de background.

El procedimiento de este método se muestra en la *Figura 8*:

- Inicialmente, se toma como background la media de las n primeras nubes de puntos. Seguidamente, se detecta la siguiente nube de puntos y se comprueba si se verifica la desigualdad (10) para determinar si el pixel es foreground (objeto) o background.

- Si un pixel F_t se clasifica como foreground entonces se ignora en el modelo de background (B= B). Este pixel F_t se almacena como pixel objeto ($O_t = F_t$).
- Si un pixel F_t se clasifica como background, entonces el background se actualiza como en (9).

El inconveniente de este método es que necesita almacenar n nubes de puntos, es decir, consume más memoria.

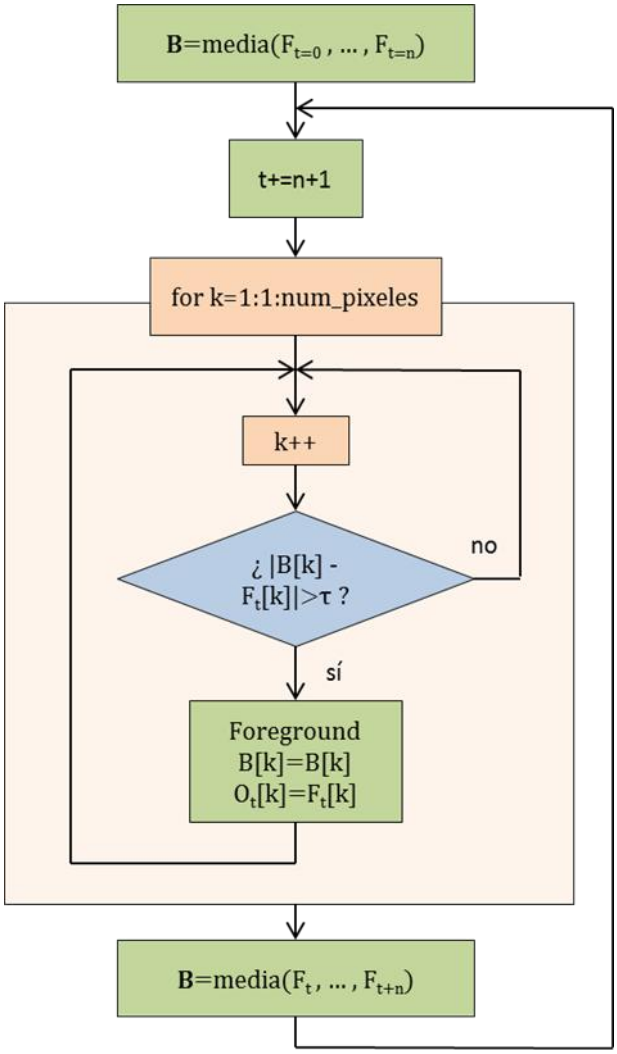

Figura 8: Diagrama del método media de n píxeles

3.1.3. Mediana de n píxeles

Este método es similar al anterior, pero en vez de calcular la media de los pixeles en n instantes de tiempo, se calcula la mediana. Esto es, el background se actualiza mediante la ecuación:

$$B_t = mediana(F_t, F_{t+1}, \ldots, F_{t+n}) \quad (11)$$

3.2. Running Average

El background se actualiza como:

$$B_t = \alpha F_t + (1 - \alpha) B_{t-1} \tag{12}$$

donde F_t y B_t son los valores de los pixeles en el instante t y la imagen de background, respectivamente; α es el radio de actualización ("updating rate") cuyo valor es pequeño para prevenir "colas" artificiales que se forman detrás de los objetos en movimiento. Su valor típico es 0,05.

Cada cierto tiempo t, cada valor de los pixeles F_t se clasifica como pixel foreground si cumple la desigualdad:

$$|B_t - F_t| > \tau \tag{13}$$

En otro caso, F_t se clasifica como pixel de background.

La *Figura 9* muestra el procedimiento de este método:

- Si un pixel F_t se clasifica como foreground entonces se ignora en el modelo de background, el cual se actualiza como $B_t = B_{t-1}$. De esta manera, se evita que el modelo de background se contamine con un pixel F_t que no pertenece a la escena de background. Este pixel F_t se almacena como pixel objeto ($O_t = F_t$).
- Si un pixel F_t se clasifica como background, entonces el background se actualiza como en la ecuación (12).

Este método se considera rápido y solo requiere almacenar los pixeles correspondientes al background (es decir, solo requiere como memoria el tamaño del background).

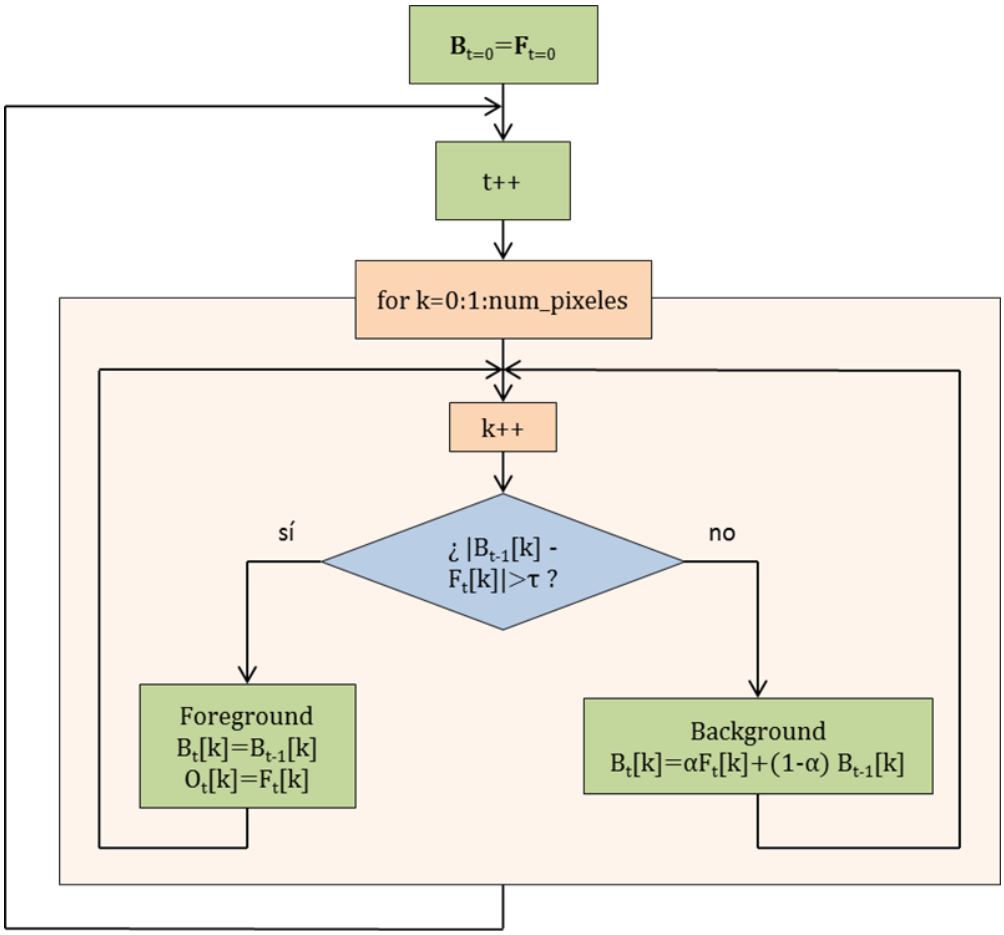

Figura 9: Diagrama del método Runningaverage

3.3. Running Gaussian Average

En este método se modela el background independientemente de la localización (x,y) de cada pixel. Se basa en el ajuste ideal de la función de densidad de probabilidad (pdf) Gaussiana con media μ_t y desviación estándar σ_t tales que:

$$\mu_{t+1} = \alpha F_t + (1-\alpha)\mu_t \tag{14}$$

$$\sigma^2{}_{t+1} = \alpha(F_t - \mu_t)^2 + (1-\alpha)\sigma^2{}_t \tag{15}$$

donde F_t son los valores de los pixeles actuales y α el peso empírico.

Ambas expresiones proporcionan la función densidad de probabilidad del background.

Cada cierto tiempo t, cada valor de los pixeles F_t se clasifica como pixel foreground si cumple la desigualdad:

$$|F_t - \mu_t| > k\sigma_t \tag{16}$$

En otro caso, F_t se clasifica como pixel de background.

3.4. Mezcla de gaussianas

Con este método cada pixel se modela de forma independiente por una mezcla de K Gaussianas ($\mu_i, \sigma_i, \omega_i$). La probabilidad de observar el valor de un determinado pixel, x, en el instante de tiempo t mediante una mezcla de gaussianas es:

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t} \eta\left(x_t - \mu_{i,t}, \sum_{i,t}\right) \quad (17)$$

donde $\sum_i$ es la matriz de covarianzas y ω_i la amplitud de pico. Se considera que cada una de las K gaussianas describe sólo uno de los objetos observables de background o foreground. En la práctica, K suele tomar un valor comprendido entre 3 y 5.

Las gaussianas son multivariables para describir los valores en los canales RGB (rojo, verde y azul). Si se asume que estos valores son independientes, se simplifica la matriz de covarianza a una matriz diagonal. Además, si se asume lo mismo para la desviación típica de los tres canales, se simplifica todavía más: ${\sigma^2}_i I$.

Para convertir (17) en un único modelo de background, es necesario un criterio que discrimine entre las distribuciones de foreground y background. Para ello, todas las distribuciones se clasifican en base a la relación entre la amplitud de pico (ω_i) y la desviación típica (σ_i). Se asume que cuanto mayor y más compacta sea la distribución, mayor probabilidad tiene de pertenecer al background. De esta manera, las primeras B distribuciones que cumplan lo anterior y verifiquen la desigualdad (18), se aceptan como background.

$$\sum_{i=1}^{B} \omega_i > T \quad (18)$$

donde T es un umbral definido.

En cada nuevo instante de tiempo t, hay que resolver simultáneamente dos problemas: asignar un nuevo valor, x_t, a la distribución que mejor se ajusta y estimar los parámetros para actualizar el modelo. De todas las distribuciones que satisfacen la desigualdad:

$$\left|x_t - \mu_{i,t}\right| > 2.5\sigma_{i,t} \quad (19)$$

la primera de ellas es la que mejor se ajusta a x_t. Por otra parte, los parámetros ($\mu_{i,t}, \sigma_{i,t}, \omega_{i,t}$) sólo se actualizan en la distribución de mejor ajuste y mediante el uso de promedios acumulativos simples como en (14) y (15). Si no se encuentra ninguna coincidencia, la última distribución se sustituye por una nueva centrada en x_t con peso (ω_i) bajo y varianza alta (σ_i).

3.5. Estimación de densidad del kernel

Se puede dar una aproximación de la función de densidad de probabilidad del background gracias al histograma de los n últimos valores de background.

La función densidad de probabilidad (pdf) puede expresarse como la suma de los kernel gaussianos centrados en los n valores más recientes del background (x_i):

$$P(x_t) = \frac{1}{n}\sum_{i=1}^{n}\eta(x_t - x_i, \sum_t) \quad (20)$$

Este modelo parece una suma de gaussianas como las de la ecuación (17). Sin embargo, hay diferencias importantes: en (17) cada gaussiana describe un "modo" principal de la función densidad de probabilidad y se actualiza conforme transcurre el tiempo. Por contra, en este método (20), cada gaussiana describe solo uno de los datos, siendo n del orden de 100 y $\sum_t$ la misma para todos los kernel. Si no se conocen los valores de background, puede usarse en su lugar los datos no clasificados; la inexactitud inicial se irá reduciendo conforme se vaya actualizando el modelo. Basándose en (20), se clasificará un pixel x_t como background si se verifica:

$$P(x_t) < \tau \quad (21)$$

La actualización del modelo se consigue actualizando el búfer de los valores de background en orden FIFO (primero en llegar, primero en salir) mediante actualización selectiva (ver el apartado 3.3). De esta manera, se evita la contaminación del modelo con valores de foreground. Sin embargo, la estimación del modelo completo requiere también estimar $\sum_t$ (asumiendo que es diagonal por simplicidad). Este es el problema clave de este método. En (20), la varianza se estima en el dominio del tiempo analizando las diferencias entre dos valores consecutivos.

3.6. Estimación basada en la técnica mean-shift

La estimación mean-shift es una técnica efectiva de gradiente de ascenso capaz de detectar los modos principales de una distribución multimodal junto con sus matrices de covarianza. Es iterativo y su número de pasos decrece con la convergencia.

El vector mean-shift se define como:

$$m(x) = \frac{\sum_{i=1}^{n} x_i g\left((x-x_i/h)^2\right)}{\sum_{i=1}^{n} g((x-x_i/h)^2)} - x \quad (22)$$

El problema de este método es que su implementación iterativa es demasiado lenta y requiere bastante memoria, en concreto, $n * size(nube\_de\_puntos)$. Esto se puede solucionar de dos maneras: con una implementación computacional o bien usando este método para detectar únicamente los modos de la función de densidad de probabilidad de background al inicio, usando después un método computacional más ligero.

3.7. Aproximación secuencial de densidad del kernel

Las técnicas vectoriales mean-shift se están empleando recientemente para problemas de reconocimiento de patrones tales como segmentación de imágenes y rastreo. Sin embargo, el

coste computacional de este método es bastante alto ya que es una técnica iterativa y requiere un estudio de la convergencia del conjunto de datos completo. Como tal, no se puede aplicar inmediatamente para modelar las funciones de densidad de probabilidad de background a nivel de pixel.

Tal y como se vio en el apartado 3.6, para solucionar los problemas de lentitud y memoria de dicho método, se utiliza la técnica mean-shift para detectar los modos de las muestras únicamente al principio. Después, los modos se propagan adaptándolos con las nuevas muestras:

$$P(x) = \alpha(modo\_nuevo) + (1-\alpha)(\sum modos\_existentes) \qquad (23)$$

La actualización a tiempo real del modelo se consigue con procedimientos heurísticos que hacen frente a los modos de adaptación, creación y fusión.

Este método es una aproximación del visto en el apartado 3.5 que proporciona prácticamente la misma exactitud pero reduce los requisitos de memoria en un orden de magnitud (y por tanto es más rápido).

3.8. Coocurrencias de variaciones en la imagen

El principal argumento de este método es que los bloques de píxeles vecinos que pertenecen al background deben experimentar variaciones similares en el tiempo. Puede resumirse como sigue:

En vez de trabajar a nivel de pixel, lo hace en bloques de NxN píxeles tratados como un vector de N^2 componentes.

APRENDIZAJE

Para cada bloque se adquiere un determinado número de muestras. Primero se calcula el promedio temporal y se denomina "imagen de variaciones" a las diferencias entre las muestras y ese promedio.

Se calcula la matriz N^2xN^2 de covarianzas con respecto al promedio anterior y se aplica una transformación del autovector. De esta forma se reduce las dimensiones de la imagen de variaciones de N^2 a K.

CLASIFICACIÓN DEL BLOQUE ACTUAL

Se considera un bloque vecino, u, con sus actuales valores de entrada y se calcula su correspondiente imagen de variaciones, llamada z_u. Se encuentran los L vecinos más cercanos a z_u en el espacio, $z_{(u,i)}$ y se expresa z_u como su interpolación lineal.

Se aplica la misma interpolación de coeficientes a los valores del bloque actual, b, los cuales ocurren al mismo tiempo que los de $z_{(u,i)}$. Esto proporciona una estimación, ${z^*}_b$, de la actual imagen de variaciones z_b.

La clave de este enfoque es que z_b y z^*_b deben estar cercanos cuando b es un bloque de background. Para medir esta cercanía se utiliza una probabilidad acumulativa entre los bloques de 8 vecinos.

3.9. Autobackgrounds ("Eigenbackgrounds")

Se basa en una descomposición de autovalores, pero esta vez aplicado a la imagen completa y no por bloques como el apartado 3.8. Esta extensión del dominio espacial permite explorar ampliamente la correlación espacial y evitar el efecto mosaico debido a las particiones en bloques.

El método puede resumirse como sigue:

APRENDIZAJE

Se adquieren muestras de n imágenes y se almacenan como columnas en una matriz A. Cada una de las n imágenes está formada por p píxeles. A continuación se calcula la media (promedio) de la imagen, μ_b, y se le resta a las imágenes.

Se calcula la matriz de covarianzas, $C = AA^T$, y se almacenan los mejores M autovectores (autobackgrounds) en una matriz de autovectores, ϕ_{Mb}, de tamaño M x p.

CLASIFICACIÓN DEL BLOQUE ACTUAL

Por cada nueva imagen, I, disponible, se proyecta sobre el autoespacio como

$$I' = \phi_{Mb}(I - \mu_b) \tag{24}$$

A continuación se proyecta I' sobre el espacio imagen como

$$I'' = \phi^T{}_{Mb} I' + \mu_b \tag{25}$$

Como el autoespacio es un buen modelo para las partes estáticas de la escena pero no para objetos pequeños en movimiento, I'' no contendrá ninguno de estos objetos.

Se clasificarán como puntos de foreground aquellos que verifiquen la desigualdad:

$$|I - I''| > T \tag{26}$$