* La resta de fondo (BS) es una técnica común y ampliamente utilizada para generar una máscara de primer plano (es decir, una imagen binaria que contiene los píxeles que pertenecen a objetos en movimiento en la escena) mediante el uso de cámaras estáticas.
* Como su nombre indica, BS calcula la máscara de primer plano realizando una resta entre el fotograma actual y un modelo de fondo, que contiene la parte estática de la escena o, más en general, todo lo que puede considerarse como fondo dadas las características de la escena observada.

El modelado en segundo plano consta de dos pasos principales:

* Inicialización de fondo;
* Actualización de antecedentes.

En el primer paso se calcula un modelo inicial del fondo, mientras que en el segundo paso ese modelo se actualiza para adaptarse a posibles cambios en la escena.

from \_\_future\_\_ import print\_function  
import cv2 as cv  
## [create]  
  
#Un objeto VideoCapture() se utiliza para leer la secuencia de vídeo de entrada o de imágenes de entrada.  
capture = cv.VideoCapture("resources/vtest.avi")  
  
while True:  
 ret, frame = capture.read()  
 if frame is None:  
 break  
  
 ## [apply]  
 #Cada fotograma se utiliza tanto para calcular la máscara de primer plano como para actualizar el fondo.  
 #Si desea cambiar la tasa de aprendizaje utilizada para actualizar el modelo en segundo plano, es posible establecer una tasa de aprendizaje específica pasando un parámetro al applymétodo.  
 ## [apply]  
  
 ## [display\_frame\_number]  
 #El número de fotograma actual se puede extraer del objeto VideoCapture y estampar en la esquina superior izquierda del fotograma actual. Se utiliza un rectángulo blanco para resaltar el número de fotograma de color negro.  
 cv.rectangle(frame, (10, 2), (100,20), (255,255,255), -1)  
 cv.putText(frame, str(capture.get(cv.CAP\_PROP\_POS\_FRAMES)), (15, 15),  
 cv.FONT\_HERSHEY\_SIMPLEX, 0.5 , (0,0,0))  
 ## [display\_frame\_number]  
  
 ## [show]  
 #Estamos listos para mostrar el cuadro de entrada actual y los resultados.  
 cv.imshow('Frame', frame)  
 ## [show]  
  
 keyboard = cv.waitKey(30)  
 if keyboard == 'q' or keyboard == 27:  
 break