Sustracción del fondo en secuencias de vídeo

FSIV - Universidad de Córdoba.

Objetivos

- Aprender a obtener un modelo de escena para separar (segmentar) objetos en movimiento.
- Para aprender a usar las interfaces que OpenCV proporciona para la entrada / salida de video en vivo (desde una cámara web) o desde archivos: cv :: VideoCapture, cv :: VideoWriter.
- Para validar su código, use los videos proporcionados en Moodle.

Descripción

En esta práctica vamos a desarrollar dos técnicas para separar el fondo del primer plano en una secuencia de vídeo. Ambas técnicas se basan en la idea de aprender un modelo de la escena estática (Background) y después calcular que hay distinto en las imágenes de la escena en momentos posteriores. Lo distinto es lo que consideramos como "primer plano" (Foreground).

Técnica 1 (básica).

La técnica básica consiste en considerar justo la imagen anterior como un modelo de la escena estática y todo lo que cambie en la imagen actual se considera que es el primer plano que se mueve.

Esta técnica simplemente detecta objetos en movimiento al umbralizar la diferencia en valor absoluto entre la imagen anterior I(t-1) y la actual I(t) de la secuencia de vídeo. Asumiendo que las imágenes están en niveles de gris aplicamos la operación:

$$M=| I(t-1) - I(t) |>= U,$$

Donde M será la máscara resultado (segmentación) y U el número de niveles mínimo de niveles de gris distintos para considerar que un pixel ha cambiado. Dependiendo del valor U podremos detectar más o menos partes de la escena en movimiento.

Esta técnica es muy sensible al ruido por lo que se suele realizar un promedio gaussiano previo de las imágenes. Además el resultado de la segmentación también mostrará "ruido de segmentación" que puede ser atenuado aplicando una operación morfológica consistente en M' = Opening(Closing(M)) usando un elemento estructural cuadrado (Figura 1).

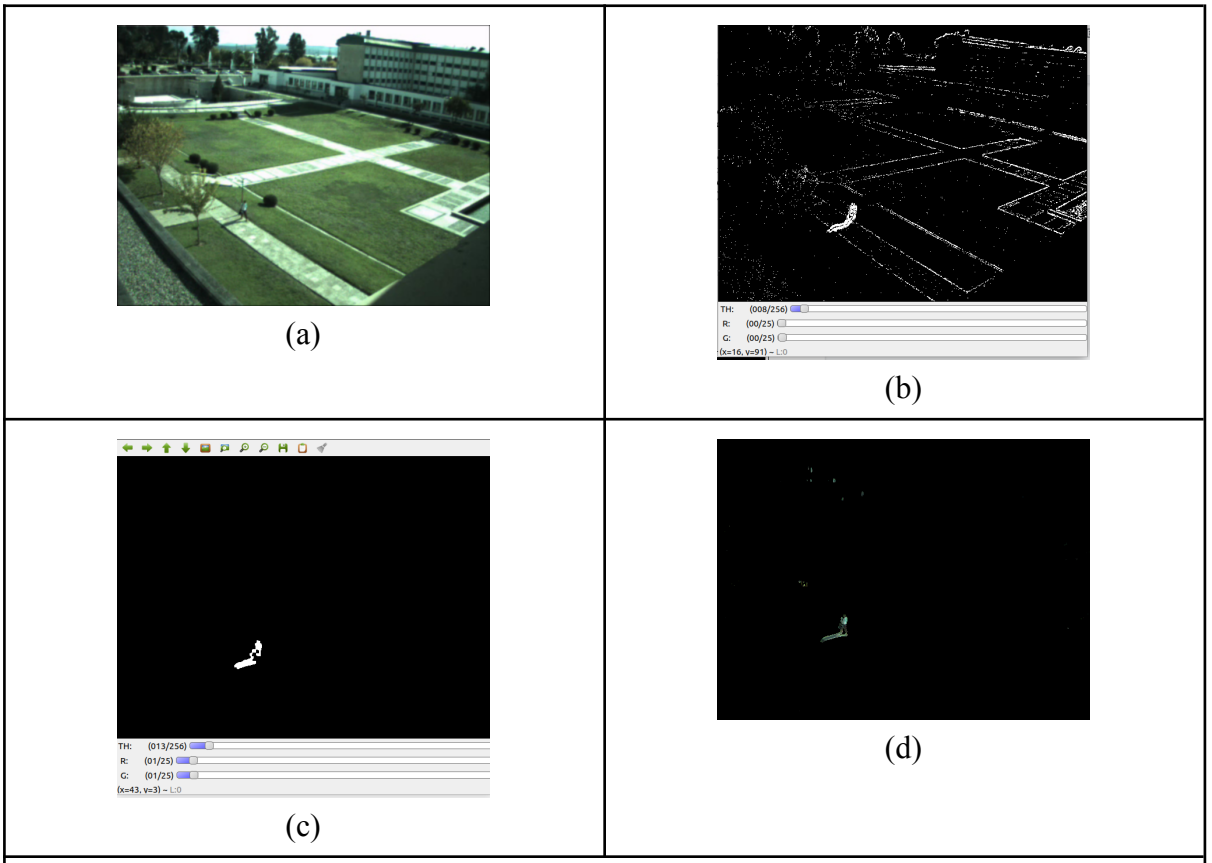

Figura 1. Ejemplo del método básico. a) Frame actual, b) segmentación sin filtrar ruido. c) segmentación usando promedio gaussiano con radio 1 y un eliminación de ruido de segmentación con ste=1. d) Imagen enmascarada obtenida al aplicar la máscara (c) a la imagen de entrada (a).

El programa debe generar un vídeo con la segmentación. El programa termina si se pulsa la tecla "ESC" o si la entrada en un vídeo al acabarse éste.

Técnica 2 (avanzada).

La técnica anterior falla si no hay movimiento en la escena y además es muy sensible al ruido. Una técnica más avanzada consiste en generar un modelo de la escena estática y usar este modelo para detectar si hay algo "nuevo".

Para modelar la escena estática consideramos que el color de un pixel se puede modelar mediante una distribución gaussiana. Para simplificar, vamos a considerar que cada canal RGB es independiente por lo que estimaremos la media y la varianza de cada canal RGB para cada pixel de la imagen, es decir, el modelo será la imagen "media" y la imagen "varianza".

Para estimar el modelo usaremos un número N dado de imágenes iniciales de la fuente de vídeo. Asumimos que durante este tiempo solo se presenta la escena a aprender sin ningún objeto en movimiento.

Una vez aprendido el modelo podemos aplicarlo en una nueva imágen de la escena para detectar lo "nuevo" de la siguiente forma:

$$|I\_m - I(t)| >= k*sqrt(I\_var),$$

siendo I_m la imagen "media" e I_var la imagen "varianza" que definen el modelo. Como las imágenes son RGB la máscara obtenida tendrá tres canales siendo la máscara final el OR de las máscaras obtenidas para cada canal.

El parámetro "k" controla la sensibilidad del método. Valores altos de k aseguran que lo detectado realmente es nuevo mientras que valores bajos aseguran que no queda nada nuevo sin detectar.

El modelo debe actualizarse para que se adapte a los cambios temporales en la escena (por ejemplo cambios de luz o incluso cambios estructurales como que alguien que se deja una mochila). Tendremos dos métodos de actualización:

- Actualización corta (por ejemplo cada 10 frames). Consiste en actualizar el modelo pero sólo de los píxeles del background, es decir de la escena estática. Para ello podemos utilizar el NOT(máscara) para realizar la actualización. Esta actualización permitirá actualizar el modelo a cambios suaves en la iluminación.
- Actualización larga (por ejemplo cada 100 frames). Consiste en actualizar el modelo de todos los píxeles sin tener en cuenta si pertenecen al background o al foreground, es decir, sin usar máscara. Esta actualización permitirá adaptarse a cambios estructurales en la escena.

Para actualizar el modelo usaremos un "running average" con factor "alfa".

$$I\_m = (1.0\text{-}alfa)I\_m + alfa * I(t),$$

$$I\_var = (1.0\text{-}alfa)I\_var + alfa* I(t)^2,$$

siendo I(t) la frame en tiempo t de la secuencia.

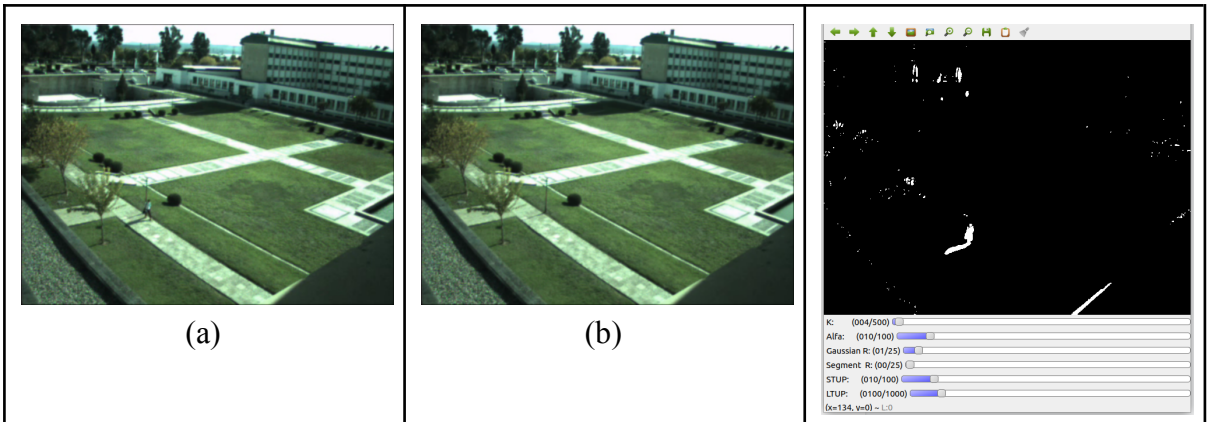

Figura 2. Ejemplo del método avanzado. a) Frame actual, b) Modelo aprendido durante las 100 primeras frames, con promedio gaussiano con radio 1. c) Segmentación (máscara) con k=0.05 y alfa=0.1. Se usa promedio gaussiano con radio 1 y eliminación de ruido de segmentación con ste=1. El periodo de actualización corto es 10 y el largo de 100.

De nuevo podemos usar un promedio gaussiano para atenuar el ruido en el sensor y un filtrado morfológico (closing+opening) de la máscara para el ruido de "segmentación".

Evaluación

La evaluación de la práctica será (se debe compilar y poder usar al el programa vsegbase):

Concepto	**Puntos**
fsiv_remove_segmentation_noise	1
fsiv_segm_by_dif	2
fsiv_apply_mask	1
vsegbase.cpp	1
fsiv_learn_gaussian_model	1
fsiv_segm_by_gaussian_model	1
fsiv_update_gaussian_model	1
vsegadv.cpp	1
Los programas permiten cambiar los parámetros de forma interactiva	1

(Nota: la entrega fuera de plazo implica una penalización en la nota obtenida).

Recursos:

Para capturar de un stream de vídeo: cv::VideoCapture.

Para escribir en un video cv::VideoWriter.

Para realizar un promedio gaussiano: cv::GaussianBlur.

Para implementar el filtrado morfológico: cv::getStructuringElement, cv::morphologyEx.

Para aprender y actualizar el modelo gaussiano: cv::accumulate, cv::accumulateSquare, cv::accumulateWeighted.