

Práctica 2. Segmentación de fondo estático.

Objetivos

- Conocer una técnica de modelado de fondo estático para la segmentación de objetivos móviles en una secuencia temporal de imágenes.
- Cálculo de un modelo estadístico de representación del fondo en una escena con fondo estático en relación a la cámara en una secuencia temporal de imágenes.

Duración estimada:

2-3 horas

Material y herramientas

- Secuencia de imágenes en formato *pgm*.
- Herramienta de programación: MatLab o C++.

Descripción

En esta práctica se va implementar y comprobar el funcionamiento de un modelo de fondo adaptativo para secuencias de imágenes en escenas con fondo estático, con el fin de poder extraer las zonas de la imagen en cada momento de la secuencia que corresponden a objetivos móviles en la escena observada.

La práctica consistirá en una implementación simplificada del algoritmo de Stauffer y Grimson visto en las clases de teoría. Esta simplificación del modelo consistirá en lo siguiente:

- La secuencia de imágenes será una secuencia monocroma, es decir, cada píxel de la imagen t de la secuencia posee un nivel de gris $I_t(x, y)$; $0 \leq x < M, 0 \leq y < N$
- El modelo de fondo consistirá en una distribución Gausiana por píxel, esto es, para cada píxel el fondo almacenará dos parámetros, la media y la desviación típica $m_t(x, y)$, $\sigma_t(x, y)$; $0 \leq x < M, 0 \leq y < N$ de la Gausiana que modela los posibles valores para ese píxel en el fondo.
- El modelo del fondo se inicializará tomando como media de las Gausianas los valores de nivel de gris de la primera imagen de la secuencia, y como desviación típica de cada píxel el valor 5, es decir,

$$m_0(x, y) = I_0(x, y), \quad \sigma_0^2(x, y) = 25; \quad 0 \leq x < M, 0 \leq y < N.$$

- Para la actualización de los valores de los parámetros del fondo a lo largo de la secuencia se utilizarán las expresiones:

$$m_t(x, y) = (1 - \rho) m_{t-1}(x, y) + \rho I_{t-1}(x, y)$$

$$\sigma_t^2(x, y) = (1 - \rho) \sigma_{t-1}^2(x, y) + \rho (I_{t-1}(x, y) - m_{t-1}(x, y))^2$$

- El parámetro de recuerdo ρ se fijará en un valor entre 0 y 1. Se pueden probar varios valores menores de 0.5.
- Para comparar cada pixel de la imagen actual con los valores de los parámetros del fondo que se tienen en ese momento, se utilizará la distancia de Mahalanobis de la siguiente forma

$$d = \sqrt{(I_t(x, y) - m_{t-1}(x, y))^2 / \sigma_{t-1}(x, y)^2}$$

- Si la distancia $d \leq U$ un umbral prefijado U , entonces se considerará que el pixel de la imagen pertenece al fondo, y habrá que actualizar los parámetros del fondo, sino, se considerará un pixel de un objetivo móvil. Probar con distintos valores del umbral hasta ajustarlo adecuadamente.

Ejercicios

Para realizar los ejercicios, utilizaremos la secuencia de 101 imágenes proporcionadas con el paquete de la práctica, nombradas como *walk470.pgm* a *walk570.pgm*. En la Figura 1 se muestra la primera imagen de la secuencia, correspondiente al vestíbulo de un edificio en el que aparece una persona andando que atraviesa el vestíbulo a lo largo de la secuencia.



Figura 1. Primera imagen de la secuencia *walk470.pgm* utilizada para la práctica.

El ejercicio consistirá en implementar el algoritmo descrito en la introducción. Los resultados del algoritmo se mostrarán por medio de imagen binaria $B_t(x, y)$; $0 \leq x < M$, $0 \leq y < N$, asignando un nivel de gris 0 a los píxeles del fondo y 255 a los píxeles de los objetos móviles detectados.

Se deberá mostrar la secuencia resultado de las imágenes binarias en pantalla, junto con la secuencia original, con el fin de comparar las regiones extraídas con la imagen original en cada momento de la secuencia.

NOTAS:

Si se programa en MatLab, se pueden utilizar las siguientes funciones:

- `imread` para leer imágenes. Ejemplo

```
I=imread('imagen.pgm','pgm');
```

donde se lee la imagen *imagen.pgm* en el array I.

- `movie` para realizar una animación. Ejemplo:

```
F(1)=im2frame(I1,gray(256));  
F(2)=im2frame(I2,gray(256));  
movie(F,10,2);
```

en la que se realizará una animación de los frames F(1) y F(2) repitiéndola 10 veces a 2 frames por segundo.