Segmentación de Fondo
Pixel-Based Adaptive Segmenter

Integrantes:
- Alvarado Canté Alondra Isabel
- Gómez Manzanero Erick Gilberto
- Múñoz Marrufo Rodrigo Hernán
- Peña González Michell Geraldin
- Zapata Pantoja Mauro José

Introducción a la segmentación de fondos

Para separar objetos en movimiento (**Foreground, FG**) de un fondo estático o dinámico (**Background, BG**).

Ya que estos pueden ser usados en:

- ➔ Seguridad y videovigilancia
- ➔ Reconocimiento facial y de marcha
- ➔ Conteo de personas y vehículos
- ➔ Aplicaciones interactivas en tiempo real

Varios obstáculos que pueden haber serian:

- ➔ Cambios en la iluminación
- ➔ Fondos con movimiento (agua, hojas, multitudes)
- ➔ Presencia de sombras
- ➔ Objetos que aparecen/desaparecen intermitentemente

¿Cómo se calcula el modelo del fondo?

PBAS es un método no paramétrico, no asume una distribución estadística como GMM.

Cada píxel se modela con un historial de valores recientes:

$$\boldsymbol{B}(x_i) = \{B_1(x_i), \ldots, B_k(x_i), \ldots, B_N(x_i)\}$$

Donde el historial almacena **N** observaciones anteriores del mismo píxel, donde típicamente **N = 35** que sirve para la mayoría de escenarios.

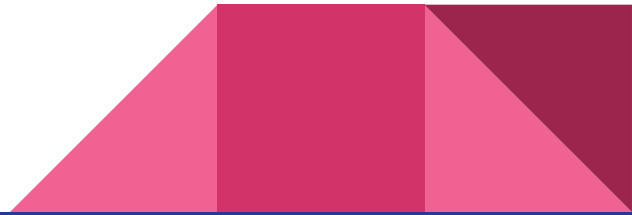
Así, el modelo del fondo se construye directamente a partir de los datos recientes, lo que permite:

- ➔ Capturar fluctuaciones naturales del fondo.
- ➔ Adaptarse a entornos donde no existe un “fondo vacío” inicial.
- ➔ Manejar transiciones suaves como atardeceres o encendidos de luces.

Cada observación no solo guarda el valor de intensidad (RGB), sino también el gradiente local pasa a ser mayor discriminación entre bordes y texturas.

Esto hace que el modelo de fondo sea más sensible a contornos de objetos y no solo a variaciones de color.

La ventaja de esto mismo conlleva a una mayor robustez en escenas donde un objeto tiene colores similares al fondo.



Estimación FG vs BG

El valor que se tiene actual $I(x_i)$ se compara con el historial del píxel de $B(x_i)$

Ya que si $I(x_i)$ es similar a suficientes muestras del historial (> #min) dentro de un umbral dinámico $R(x_i)$ se clasifica como el fondo. Si este no cumple la condición, se pasa a clasificar como un FG.

Lo tenemos como: $$F(x_i) = \begin{cases} 1 & \#\{dist(I(x_i), B_k(x_i)) < R(x_i)\} < \#_{min} \\ 0 & \text{else} \end{cases}$$

Algunos factores clave en esta estimación, tenemos que:

Su umbral dinámico $R(x_i)$

- ➔ Cada pixel tiene un umbral distinto, donde es ajustado según su variabilidad.
- ➔ Cada área estática tiende a tener un umbral más bajo (siendo más sensible)
- ➔ Las áreas dinámicas tienen un umbral más alto (son menos sensibles a falsos positivos)

El número mínimo de coincidencias define cuántos valores del historial deben ser similares al valor actual, esto evita que una sola coincidencia engañe al modelo.

Actualización del modelo de fondo

Tenemos que el modelo debe estarse actualizando constantemente para poder adaptarse a cambios.

- ➔ Solo se actualiza si el píxel fue clasificado como fondo.
- ➔ Se reemplaza aleatoriamente un valor del historial como el nuevo valor de $I(x_i)$
- ➔ La probabilidad de actualización depende del parámetro $T(x_i)$ que define la tasa de actualización, donde es hecho por:

$$p = 1/\bar{T}(x_i)$$

Cada actualización se hace de la siguiente manera que:
La tasa de $T(x_i)$ controla la velocidad de actualización, donde:

- ➔ Si el pixel es fondo, $T(x_i)$ disminuye, lo que hace que haya más actualizaciones
- ➔ Si es un primer plano, $T(x_i)$ aumenta, lo que conlleva a menos actualizaciones

También se puede actualizar un píxel vecino, lo que permite absorber bordes de objetos y eliminar falsos positivos, lo que resulta en que los objetos erróneos o el ruido desaparecerán con el tiempo.

$R(x_i)$ es el umbral de decisión se regula con base en la variabilidad promedio del píxel.

$T(x_i)$ es la tasa de aprendizaje cambia con el tiempo para evitar que los objetos en movimiento lento se conviertan rápidamente en fondo.

Ambos parámetros tienen límites superiores e inferiores, lo que evita errores extremos, esto hace que PBAS sea autoajustable por píxel y por tiempo, y mucho más flexible que métodos con parámetros fijos.

Problemas de escena

PBAS fue diseñado para ser robusto ante condiciones complejas:

- ➔ Cambios de iluminación
  - ◆ Puede adaptarse a variaciones graduales (amanecer, luces encendidas).
  - ◆ El modelo se ajusta lentamente sin clasificar erróneamente todo como FG.

- ➔ Sombras
  - ◆ Aunque no implementa un modelo explícito, su actualización adaptativa logra buen desempeño contra sombras.
  - ◆ Evita que sombras se consideren falsos objetos.

- ➔ Objetos del fondo en movimiento
  - ◆ Los objetos como agua, ramas de árboles, etc; son atacadas por el umbral $R(x_i)$ aumenta en estas áreas, lo que reduce falsas detecciones.

- ➔ Objetos intermitentes o estáticos temporales
  - ◆ Las personas que se detienen o maletas que se dejan en el suelo, se tiene que el parámetro $T(x_i)$ regula la incorporación al fondo:
    - Objetos grandes y estáticos se mantienen como FG más tiempo.
    - Pequeños falsos positivos se absorben rápido.

- ➔ Ruido de cámara y jitter
  - ◆ Resistente a vibraciones leves gracias al ajuste dinámico.

Conclusión

Con toda esta información, podemos concluir que PBAS es un modelo de fondo no paramétrico con controladores adaptativos, ya que tiene las ventajas de que se ajuste de forma local y dinámica sus parámetros, sea robusto a condiciones reales como la iluminación, sombras y objetos dinámicos, y tiene una menor dependencia de parámetros globales fijos.

Visualización de un ejemplo

https://youtu.be/DjCLUQg4dow

Explicación del pseudocódigo

```
# Parámetros iniciales
N = número_de_muestras_por_píxel
α = tasa_adaptación_umbral
β = tasa_adaptación_learning_rate

Para cada píxel (x,y):
    Inicializar memoria S(x,y) con N muestras
    Inicializar R(x,y) = umbral_inicial
    Inicializar T(x,y) = tasa_de_actualización_inicial
```

Aquí se prepara la memoria de cada píxel, el **umbral inicial** y la **tasa de actualización**.

Explicación del pseudocódigo

```
I = valor_actual_del_píxel

# Calcular distancia mínima contra la memoria
D_min = +∞
Para cada muestra S_i en S(x,y):
    d = distancia(I, S_i)
    Si d < D_min:
        D_min = d

# Decisión FG / BG
Si D_min < R(x,y):
    etiqueta(x,y) = BG   # Fondo
Sino:
    etiqueta(x,y) = FG   # Primer plano
```

Aquí está la decisión central: si la distancia mínima es menor al umbral → **BG**, si no → **FG**.

Explicación del pseudocódigo

```
Si etiqueta(x,y) == BG:
    # Promedio de distancias de las muestras válidas
    D_mean = promedio_de_distancias
    R(x,y) = R(x,y) + α * (D_mean - R(x,y))
```

```
# Actualizar memoria solo si es BG
Si etiqueta(x,y) == BG:
    Con probabilidad 1/T(x,y):
        Reemplazar una muestra de S(x,y) con I
        # Opción: actualizar también un vecino

# Ajustar tasa de actualización
Si etiqueta(x,y) == FG:
    T(x,y) = T(x,y) + β
Sino:
    T(x,y) = max(T_min, T(x,y) - β)
```

- El **umbral R(x,y)** se ajusta automáticamente:
  - Si el píxel es muy variable → R sube (menos sensible).
  - Si es estable → R baja (más sensible).

- La memoria se actualiza poco a poco para que el fondo evolucione con la escena.
- La tasa `T(x,y)` regula qué tan rápido aprende: más lento en FG, más rápido en BG.