

Maestría en Matemáticas Aplicadas

Dr. Francisco Javier Hernández

Mario Alberto Tapia de la Cruz

1. Tarea 4. Visión por Computadora

1.1. PBAS.

Hofmann et. al combinaron el ViBe junto con SACON para crear el PBAS para segmentación de foreground (primer plano).

PBAS usa un historial de N valores de imagen como el modelo de fondo, y usa un update similar al ViBe (no se hablará de ViBe aquí dado que el algoritmo elegido fue PBAS, pero al inicio se contempló hacer el informe sobre ViBe pero no me funcionaba al usar pybgs). Observación: PBAS no trata a los parámetros como parámetros fijos, pero sí como variables de estado, que pueden cambiar de forma dinámica sobre el tiempo para cada pixel.

¿Cómo definimos nuestro modelo de fondo (background)?

El modelo $B(x_i)$ está definido por un arreglo de N valores de píxel recientemente observados, de la forma:

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

¿Cómo clasificamos un pixel como fondo? Segmentaion Decision

Cada píxel x_i se clasifica como fondo de la siguiente manera:

- Si su valor de pixel $I(x_i)$ es cercano a al menos $\#_{min}$ de los N valores de fondo en términos de una cierta decisión de tolerancia $R(x_i)$. Entonces el fondo está definido como:

$$F(x_i) = \begin{cases} 0 & \text{si } \# \{dist(I(x_i), B_k(x_i)) < R(x_i)\} \geq \#_{min} \\ 1 & \text{en otro caso} \end{cases} \quad (1)$$

Cuando tenemos $F = 1$ significa primer plano (foreground).

Observación. $R(x_i)$ es definido de forma separada para cada píxel, y puede cambiar de forma dinámica, además debe ser capaz de adaptarse de forma automática a áreas con fondo muy dinámico.

Observación 2. $\#_{min}$ es un número, y es un parámetro global fijo.

Actualizamos el modelo de fondo.

Dado que las regiones en primer plano no pueden ser usadas para la actualización, el modelo de fondo se actualiza solamente para aquellos píxeles que son parte del fondo (los $F = 0$), esto significa lo siguiente.

Para un cierto índice $k \in 1, \dots, N$ (elegido de forma uniformemente aleatoria), el valor del modelo de fondo correspondiente $B_k(x_i)$ se reemplaza por el valor de píxel actual $I(x_i)$, esto permite que dicho valor de píxel sea aprendido por el fondo.

Esta actualización se hace con probabilidad $p = 1/T(x_i)$, donde $T(x_i)$ define la tasa de actualización.

¿Qué pasa con los píxeles de alrededor de x_i ?

Cuando x_i es clasificado como píxel de fondo, el fondo será actualizado y el píxel vecino y_i que podría ser un píxel de primer plano podría ser actualizado también. Esto significa que los píxeles en primer plano en las fronteras serán gradualmente incluidos en el modelo de fondo. Esto depende del parámetro de actualización T_i , de forma que:

$$T(x_i) = \begin{cases} T(x_i) + \frac{T_{inc}}{d_{min}(x_i)} & \text{si } F(x_i) = 0 \\ T(x_i) - \frac{T_{dec}}{d_{min}(x_i)} & \text{si } F(x_i) = 1 \end{cases} \quad (2)$$

T_{inc} , T_{dec} son parámetros fijos, T_{inc} significa que $T(x_i)$ se incrementa y T_{dec} significa que $T(x_i)$ decrece, además estos actúan como cotas para T , dichos valores no pueden salir de una frontera específica.

Observación. Cuando se trata de un fondo muy dinámico, $T(x_i)$ se mantiene constante o solo cambia ligeramente, esto significa que el primer plano detectado de forma errónea no permanecerá ahí por mucho tiempo.

Observación 2. En el caso de fondo estático la clasificación es más solida, entonces $T(x_i)$ se incrementa de forma más rápida, así validando el modelo de fondo y haciendo que este se actualice menos.

¿Cómo actualizamos el umbral de decisión $R(x_i)$?

En un vídeo podemos tener areas con fondos dinámicos altos (i.e agua, árboles que se mueven con el viento, etc.), y áreas con pocos cambios o cambios nulos. Por ende, nuestro umbral $R(x_i)$ debe incrementarse de manera que no incluya objetos en el primer plano, y para regiones estáticas el $R(x_i)$ debe ser bajo, de modo que pequeñas desviaciones nos lleven a una decisión para nuestro primer plano. De esta forma $R(x_i)$ debe ser capaz de adaptarse de forma automática.

Por lo que además de el conjunto $B(x_i)$, también creamos el arreglo $D(x_i) = \{D_1(x_i), \dots, D_N(x_i)\}$ de distancias mínimas de decisión. De forma que cada que se lleve a cabo una actualización de $B_k(x_i)$, la distancia mínima actualmente observada $d_{min}(x_i) = \min_k \text{dist}(I(x_i), B_k(x_i))$ sea escrita en el arreglo: $D_k(x_i) \leftarrow d_{min}(x_i)$. Además podemos calcular el promedio de estos valores de la forma $\bar{d}_{min}(x_i) = 1/N \sum_k D_k(x_i)$, y funciona como una medida de la dinámica del fondo.

De este modo $R(x_i)$ se adapta de forma dinámica como sigue:

$$R(x_i) = \begin{cases} R(x_i) \cdot (1 - R_{inc/dec}) & \text{si } R(x_i) > \bar{d}_{min}(x_i) \cdot R_{scale} \\ R(x_i) \cdot (1 + R_{inc/dec}) & \text{en otro caso} \end{cases} \quad (3)$$

donde $R_{inc/dec}$, R_{scale} son parámetros fijos, y $\bar{d}_{min}(x_i)$ es el promedio de la distancia mínima observada d_{min} .

Un incremento drástico en las dinámicas del fondo nos lleva a un incremento lento de $R(x_i)$ hacia un umbral de decisión más grande $R(x_i)$.

Evaluación de rendimiento según el paper original.

Se probó su rendimiento en seis categorías, y contra seis método, y PBAS funcionó mejor para las siguientes: baseline (poco movimiento), shadow (si confunde sombras con cosas en primer plano), thermal (imágenes capturadas con cámaras térmicas). Además al compararse con los otros métodos en el estado del arte resultó ser bastante bueno en las métricas FPR, FNR, PBC y F1, pero perdió contra GMM en Specificity y FPR contra él (este fue el único que sobrepasó a PBAS en dos categorías).

Experimentación.

Se implementaron las siguientes pruebas del modelo PBAS con la librería de **pybgs** en python, usando los siguientes parámetros vistos en la tabla, además de que se agregarán capturas de pantalla de las diferencias.

Config.	N	#min	Rinc/dec	Rlower
1 (Default)	35	2	0.05	18
2	10	1	0.1	10
3	50	3	0.01	20
4	35	2	0.05	15

Config.	Rscale	Tdec	Tinc	Tlower	Tupper
1 (Default)	5	0.05	1	2	200
2	4	0.1	0.5	1	100
3	6	0.02	2	3	300
4	5	0.01	2	2	400

Cuadro 1: Configuraciones de parámetros de PBAS para diferentes objetivos experimentales.

Y a continuación se presentarán capturas de pantalla de las 4 configuraciones de parámetros. Se puede observar que visualmente la configuración 1 (default del paper) sigue siendo mejor para separar el primer plano (foreground) con artefactos que puedan aparecer sin ser parte de un objeto en movimiento. Además la configuración 2 parece ser muy mala, para esta se decrecieron todos los parámetros y parece que hace una muy mala detección de los objetos en movimientos (foreground) y los objetos del background como lo son algunas líneas de la calle o edificios.

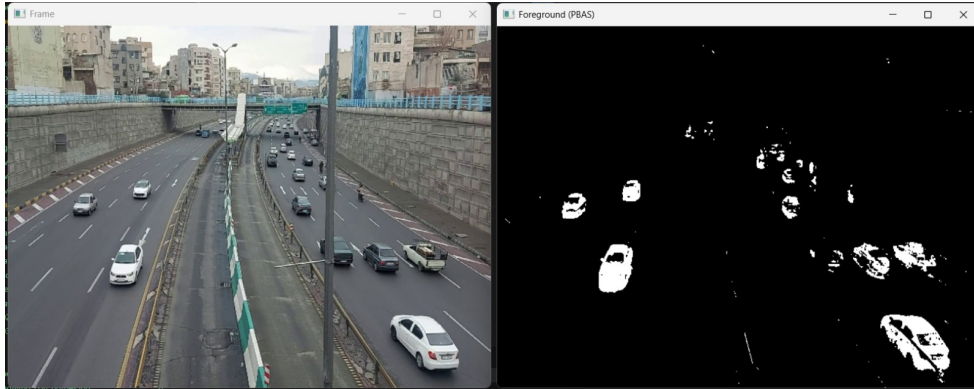


Figura 1: Configuración 1 – Default

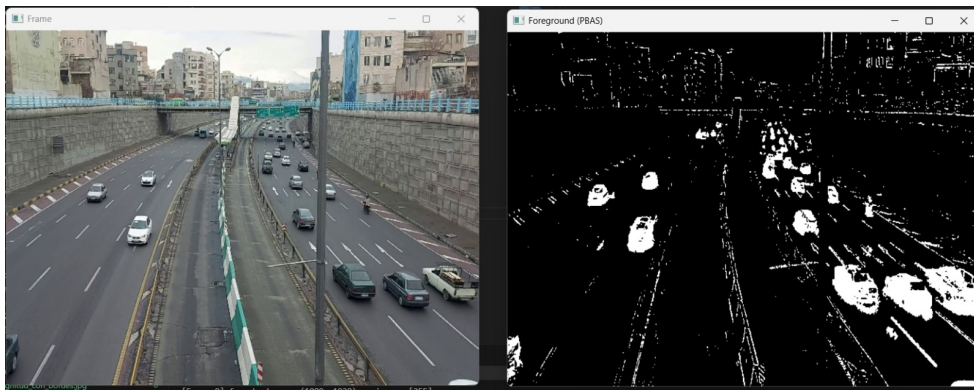


Figura 2: Configuración 2

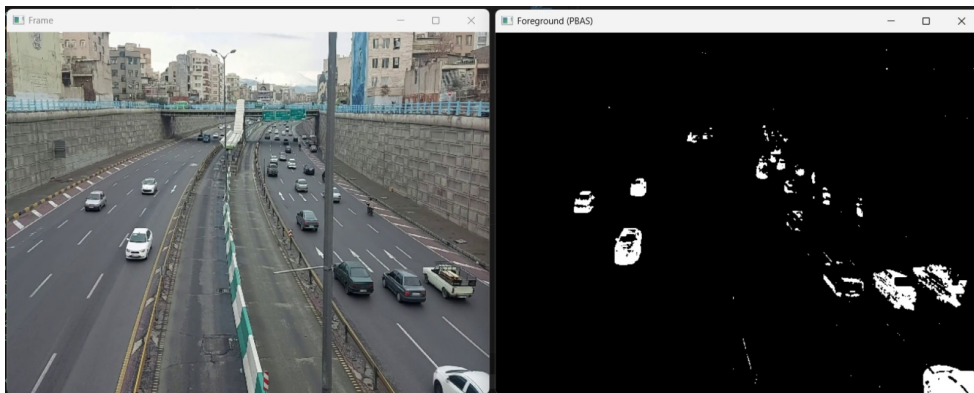


Figura 3: Configuración 3

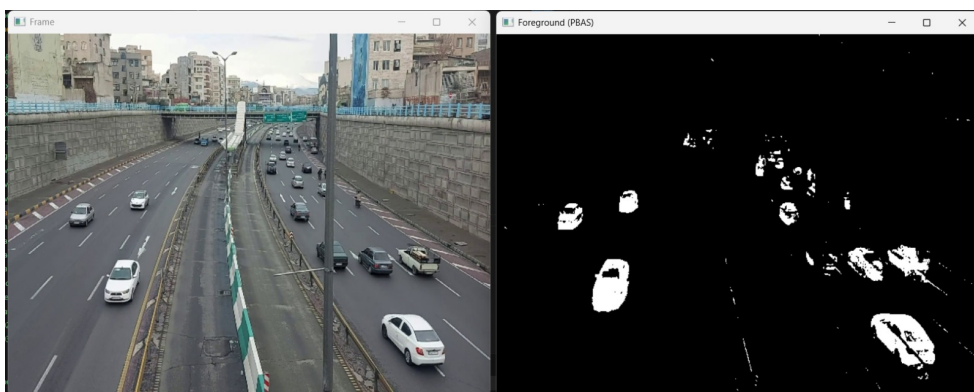


Figura 4: Configuración 4

¿Qué problemas de la escena toma en cuenta PBAS?

Gracias a $R(x_i)$ se puede permitir cambios ligeros o repentinos de brillo, pues este se ajusta de forma dinámica por píxel de acuerdo a las variaciones recientes de este mismo.

Además gracias a $D(x_i)$ y al cálculo de la media \bar{d}_{min} , el algoritmo puede detectar regiones con alta variabilidad, y en conjunto nuevamente con $R(x_i)$, hace que zonas como agua, árboles con aire no activen el primer plano de manera no eficiente.

Las cámaras con jitter son manejadas de forma eficiente al trabajar píxeles de forma independiente.

Además en el Outlook and Conclusion se menciona que no se detectan sombras de forma explícitas, por lo que podría ser algo a considerar en el futuro.

Se podría concluir que PBAS es un método de modelado de fondos muy eficiente al haber sobrepasado modelos en el estado del arte.