



Ciencia y Tecnología

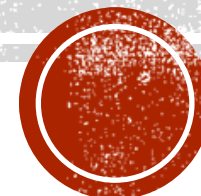
Secretaría de Ciencia, Humanidades, Tecnología e Innovación



CIMAT
UNIDAD MÉRIDA

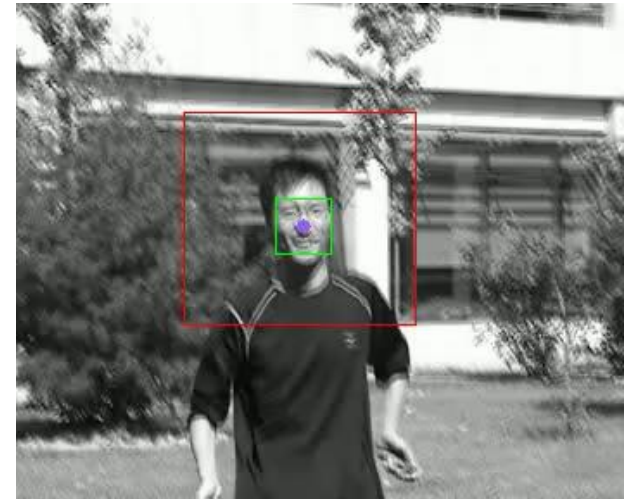
SEGUIMIENTO DE OBJ. USANDO MÉTODOS TRADICIONALES

Dr. Francisco J. Hernández López
SECIHTI – CIMAT-Mérida
fcoj23@cimat.mx, www.cimat.mx/~fcoj23



SEGUIMIENTO DE OBJETOS EN UN VIDEO

- La salida de un algoritmo de seguimiento de objetos depende de la aplicación y la representación utilizada para describir el objeto que se desea seguir, dicha salida puede ser:
 - El contorno del objeto
 - Coordenadas 2D de su centro de masa
 - Posición 3D en coordenadas del mundo
 - Postura de un objeto articulado, etc.
- Aplicaciones:
 - Interface Humano-Maquina
 - Reconocimiento de gestos
 - Estimación de pose del cuerpo y rostro
 - Análisis y reconocimiento de expresiones faciales



SEGUIMIENTO DE OBJETOS EN UN VIDEO (C1)

➤ Vigilancia Inteligente

- Monitoreo eficiente de bancos, tiendas, estacionamientos, etc.
- Clasificación de movimiento (ej. Humano, No humano)
- Reconocimiento y seguimiento de rostros para el control de accesos
- Seguimiento a través de múltiples cámaras

➤ Realidad virtual y animación por computadora

- Insertar y animar un avatar en un ambiente virtual
- Sintetizar movimiento humano para videojuegos
- Entrenamiento y análisis del rendimiento de atletas

CRITERIOS A TOMAR EN CUENTA

- Dimensionalidad del espacio de seguimiento
 - Pueden ser clasificados como 2D o 3D
 - 2D → Movimiento en el plano de la imagen
 - 3D → Estima el movimiento del objeto 3D usando la información 2D adquirida por una o varias cámaras
- Estructura del objeto a seguir
 - Afecta el tipo de movimiento que se requiere estimar
 - Estimación de objetos rígidos o deformables
 - Un movimiento articulado puede ser definido como un movimiento rígido a pedazos, donde las partes rígidas se ajustan a las restricciones del movimiento rígido

CARACTERÍSTICAS

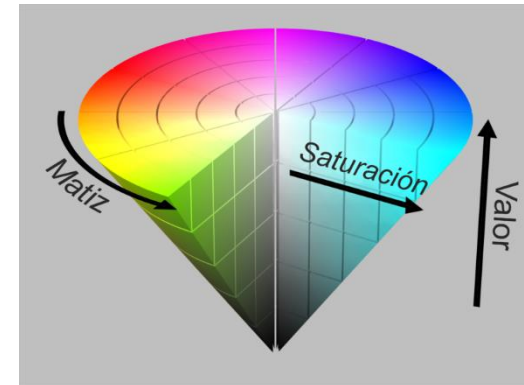
- Para realizar un seguimiento a través del tiempo, se puede considerar un emparejamiento (*matching*) de los objetos en *frames* consecutivos usando algún tipo de información:
 - Posición
 - Color
 - Velocidad
 - Textura, etc.
- Inicialización
 - Puede realizarse offline u online
 - La idea es obtener información acerca de la cámara y/o la escena y/o el objeto que deseamos seguir
 - En caso de usar substracción de fondo, puede incluir las imágenes de referencia del BG
- Manejo de oclusiones
 - Pueden ser parciales o totales
 - Self-occlusion → Partes del objeto que son ocluidas por el mismo objeto (ej. Oclusión del rostro con las manos en una conversación)
 - Un camino para atacar el problema es la Re-inicialización

SEGUIMIENTO DE OBJETOS RÍGIDOS 2D

- El objetivo es determinar el movimiento de la proyección de uno o más objetos rígidos en el plano de la imagen
- Una suposición básica es que existe solo un movimiento relativo rígido entre la cámara y la escena observada (ej. Seguimiento de autos)
- Los métodos para el seguimiento de objetos rígidos 2D pueden ser clasificados en las siguientes categorías:
 - Métodos basados en regiones
 - Métodos basados en contornos
 - Métodos basados en plantillas (*templates*)
 - Métodos basados en puntos característicos
 - Métodos basados en deep learning

SEGUIMIENTO DE OBJETOS BASADO EN REGIONES

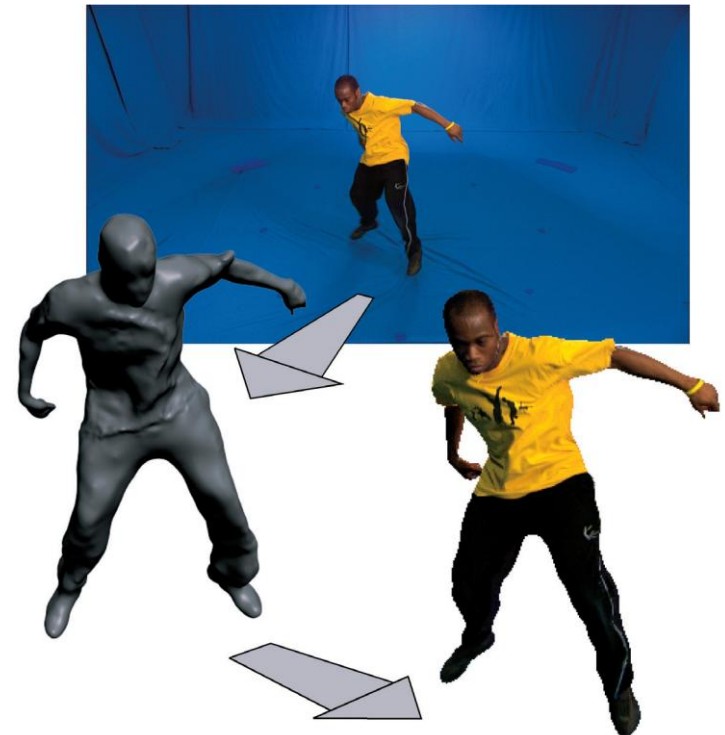
- Una región de la imagen puede ser definida como el conjunto de píxeles que tienen características homogéneas, obtenido con:
 - Segmentación
 - Caja envolvente
- Segmentación a color es la parte esencial de los algoritmos de seguimiento de objetos basados en color
- El problema principal son los cambios de iluminación
- Un camino común para intentar obtener un cierto grado de invariancia de iluminación es usando solo los valores de la crominancia en un espacio de color adecuado:
 - HS → Matiz y Saturación del espacio de color HSV
 - Espacio de color RGB normalizado



https://es.wikipedia.org/wiki/Modelo_de_color_HSV

SEGUIMIENTO DE OBJETOS BASADO EN REGIONES (C1)

- Existen algoritmos que se enfocan en el seguimiento de humanos (partes del cuerpo, manos, rostros, piel, etc.)
- Un escenario simple que permite tener un algoritmo robusto es el croma keying



<http://www.ee.surrey.ac.uk/CVSSP/VMRG/surfcap.html>

SEGUIMIENTO USANDO HISTOGRAMAS

- Inicialización

- Calcular el histograma de color de todos los objetos de interés en la escena a partir de un número de frames del video y entonces almacenar los histogramas en

$O_i^r \rightarrow$ histograma de referencia

- Para cada uno de los siguientes frames

- Calcular el histograma de color para cada objeto a seguir

$O_i^t \rightarrow$ histograma para cada objeto en cada posición candidata

- Entonces se comparan los histogramas O_i^t con los respectivos histogramas de referencia O_i^r , y el lugar en donde hacen match será la posición del objeto en el frame actual

- Un criterio para medir la similitud entre los histogramas puede ser:

$$SSD(O_i^t, O_i^r) = \sum_{n=1}^{nb} (O_{i,n}^r - O_{i,n}^t)^2 \quad \text{con } nb \rightarrow \text{número de bins}$$

SEGUIMIENTO DE OBJETOS BASADO EN KERNEL

- La representación del *target* basado en el histograma se realiza mediante un enmascaramiento espacial con un kernel isotrópico (circularmente simétrico)
- Representación:
 - Sea $\{\vec{x}_i^*\}_{i=1,\dots,n} \rightarrow$ posiciones del pixel normalizados en la región target
 - Sea $\{\vec{x}_i\}_{i=1,\dots,n_h} \rightarrow$ posiciones del pixel normalizados en la región candidata

Modelo del target

$$\hat{q} = \{\hat{q}_u\}_{u=1,\dots,m} \quad \sum_{u=1}^m \hat{q}_u = 1$$

$$\hat{q}_u = C \sum_{i=1}^n k(\|\vec{x}_i^*\|^2) \delta[b(\vec{x}_i^*) - u],$$

$$\text{con } C = \frac{1}{\sum_{i=1}^n k(\|\vec{x}_i^*\|^2)}$$

$b(\vec{x}_i) \rightarrow$ valor del pixel $[1, \dots, m]$

Modelo del candidato

$$\hat{p}(\vec{y}) = \{\hat{p}_u(\vec{y})\}_{u=1,\dots,m} \quad \sum_{u=1}^m \hat{p}_u = 1$$

$$\hat{p}_u(\vec{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right) \delta[b(\vec{x}_i) - u],$$

$$\text{con } C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right)}$$

Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking." Pattern Analysis and Machine Intelligence, IEEE Transactions on 25.5 (2003): 564-577.

Seguimiento de objetos (Met. Tradicionales). Francisco J. Hernández-López

Ago-Dic 2025

ALGORITMO DE SEGUIMIENTO DE OBJETOS BASADO EN KERNEL

- Dado el modelo del target $\{\hat{q}_u\}_{u=1,\dots,m}$ y su posición inicial \hat{y}_0 en el frame $t - 1$

1. Inicializar la posición del target en el frame actual con \hat{y}_0 , calcular $\{\hat{p}_u(\vec{y}_0)\}_{u=1,\dots,m}$ y evaluar

$$\rho[\hat{p}(\vec{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\vec{y}_0) \hat{q}_u} \quad \text{Coeficiente de Bhattacharyya}$$

2. Calcular los pesos $\{w_i\}_{i=1,\dots,n_h}$ $w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\vec{y}_0)}} \delta[b(\vec{x}_i) - u]$

3. Calcular la nueva posición del target candidato

con $g(z) = -k'(z)$ (menos la derivada del perfil del kernel)

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right)}$$

ALGORITMO DE SEGUIMIENTO DE OBJETOS BASADO EN KERNEL (C1)

4. Calcular $\{\hat{p}_u(\vec{y}_1)\}_{u=1,\dots,m}$ y evaluar

$$\rho[\hat{p}(\vec{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\vec{y}_1) \hat{q}_u}$$

Coefficiente de Bhattacharyya

5. Mientras $\rho[\hat{p}(\vec{y}_1), \hat{q}] < \rho[\hat{p}(\vec{y}_0), \hat{q}]$, hacer

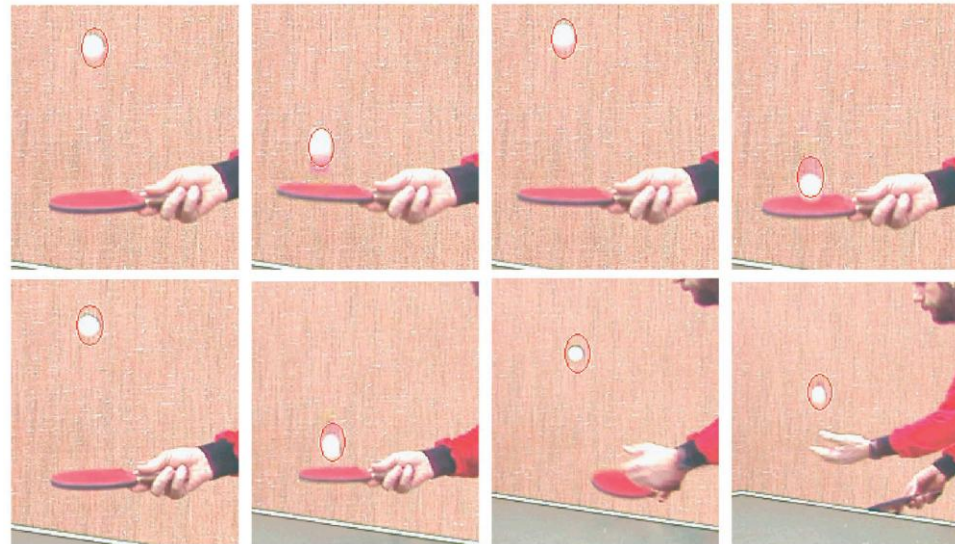
- a) Actualizar $\vec{y}_1 \leftarrow \frac{1}{2}(\vec{y}_0 + \vec{y}_1)$
- b) Evaluar $\rho[\hat{p}(\vec{y}_1), \hat{q}]$

6. Si $\|\vec{y}_1 - \vec{y}_0\|_2 < \epsilon$, entonces

- a) parar

De lo contrario

- a) Actualizar $\vec{y}_0 \leftarrow \vec{y}_1$
- b) Ir al paso 2



Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking." Pattern Analysis and Machine Intelligence, IEEE Transactions on 25.5 (2003): 564-577.

Seguimiento de objetos (Met. Tradicionales). Francisco J. Hernández-López

Ago-Dic 2025

SEGUIMIENTO DE OBJETOS BASADO EN PUNTOS CARACTERÍSTICOS

- La idea es estimar los parámetros del movimiento de puntos característicos en una secuencia de video
- Sea $A = \{A_0, A_1, \dots, A_{N-1}\}$ el conjunto de N frames y $\vec{m}_i(x_i, y_i)$, $i = 0, \dots, N - 1$, las posiciones de la característica (x_i, y_i) en los frames i
- La tarea es determinar un vector de movimiento $\vec{d}_i(d_{x,i}, d_{y,i})$, el cual determina la posición de la característica en el siguiente frame, es decir:

$$\vec{m}_{i+1} = \vec{m}_i + \vec{d}_i$$

- El objeto a seguir, comúnmente se define por los puntos característicos dentro de una caja envolvente

PUNTOS CARACTERÍSTICOS

- Shi y Tomasi, 1994, proponen lo siguiente:
 - Para cada punto característico candidato, se construye

$$Z = \begin{bmatrix} \sum_W \frac{\partial I}{\partial x} \frac{\partial I}{\partial x} & \sum_W \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \sum_W \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \sum_W \frac{\partial I}{\partial y} \frac{\partial I}{\partial y} \end{bmatrix}$$

donde W es una ventana de $n \times n$ centrada en el punto característico

- Un punto característico es aquel punto donde el eigenvalor más pequeño de Z es más grande que un umbral predefinido

ESTIMAR LOS DESPLAZAMIENTOS

- En [Tomasi y Kanade, 1991], el desplazamiento de un punto característico se elige de tal forma que minimiza la siguiente función de error

$$\epsilon = \iint_W [I_i(\vec{m} - \vec{d}) - I_{i+1}(\vec{m})]^2 w(\vec{m}) d\vec{m}$$

Donde \vec{d} es el vector de desplazamientos y $w(\vec{m})$ es una función de pesos (en el caso más simple $w(\vec{m}) = 1$)

- Cuando el desplazamiento entre las imágenes I_i y I_{i+1} es pequeño, entonces \vec{d} puede estimarse resolviendo

$$Z\vec{d} = \vec{b}$$

$$\text{con } \vec{b} = \iint_W [I_i(\vec{m}) - I_{i+1}(\vec{m})]^2 g(\vec{m}) w(\vec{m}) d\vec{m} \text{ y } g(\vec{m}) = \begin{bmatrix} \frac{\partial(I_i(\vec{m}) + I_{i+1}(\vec{m}))}{\partial x} \\ \frac{\partial(I_i(\vec{m}) + I_{i+1}(\vec{m}))}{\partial y} \end{bmatrix}$$

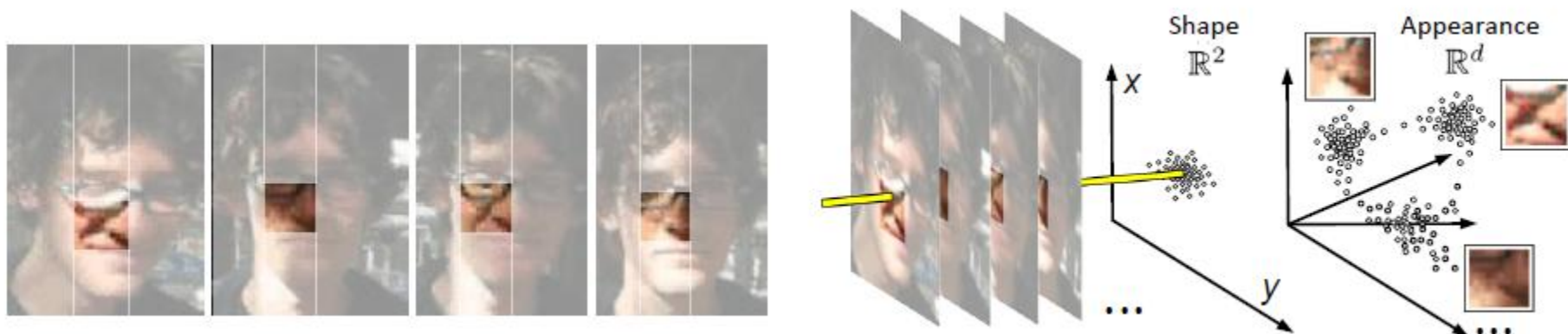
APPEARANCE LEARNING IN EVIDENTIAL NUISANCE (ALIEN)

- Aprendizaje de apariencia en factores “molestos” evidentes como:
 - Variaciones de contraste y punto de vista
 - Oclusiones
 - Cuantización del sensor
 - Cambios de iluminación
- ¿Se pueden eliminar estos factores “molestos” para que la única información restante sea la necesaria para rastrear un objeto en la escena?

Pernici, Federico, and Alberto Del Bimbo. "Object tracking by oversampling local features." *IEEE transactions on pattern analysis and machine intelligence* 36.12 (2013): 2538-2551.

ALIEN

- Consiste en un **clasificador discriminante** entre Objeto y Contexto, que evita la actualización de la plantilla del objeto durante una posible oclusión
- Se asume tener características invariantes locales para la representación del objeto (SIFT)
- Se toman múltiples instancias de las mismas características bajo diferentes condiciones (Sobremuestreo)



Alineamiento de múltiples instancias de características locales. Tomado de [Pernici, 2013]

[Pernici, 2013] Pernici, Federico, and Alberto Del Bimbo. "Object tracking by oversampling local features." *IEEE transactions on pattern analysis and machine intelligence* 36.12 (2013): 2538-2551.

DIAGRAMA A BLOQUES DE ALIEN

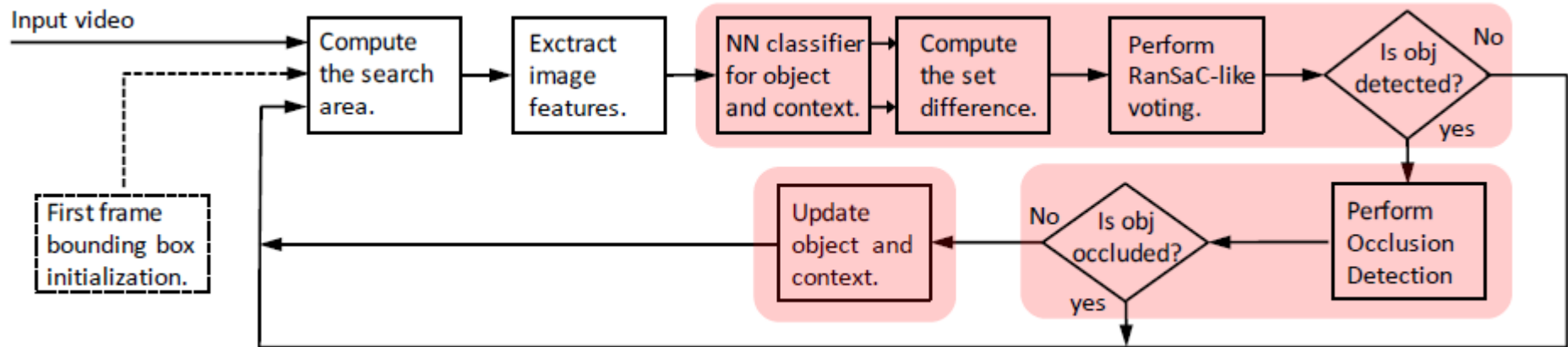


Diagrama a bloques del método ALIEN. Tomado de [Pernici, 2013]

[Pernici, 2013] Pernici, Federico, and Alberto Del Bimbo. "Object tracking by oversampling local features." *IEEE transactions on pattern analysis and machine intelligence* 36.12 (2013): 2538-2551.

ALGORITMO ALIEN

Algorithm 1: ALIEN tracking algorithm.

Input: Initial object bounding box x_0 ,

Output: Estimated Object State $\hat{x}_t = (\hat{x}_t, \hat{y}_t, \hat{\theta}_t, \hat{s}_t)$, object shape and appearance \mathcal{T}_t and object context appearance \mathcal{C}_t .

```

1 repeat
2   Crop out the search region
    $S_t = \{x \in \mathbb{R}^4 : \|x - \hat{x}_{t-1}\|_\infty < r\}$  and extract local
   features  $\mathcal{S}_t = \{(p, d) | p \in S_t, d \in \mathbb{R}^d\}$ .
3   // Compute matching features
    $\mathcal{T}_t^* = \{(p, d) \in \mathcal{S}_t | \frac{\|d - 1NN_{\mathcal{T}}(d)\|}{\|d - 2NN_{\mathcal{T}}(d)\|} < \lambda_{\mathcal{T}}\}$ ;
4    $\mathcal{C}_t^* = \{(p, d) \in \mathcal{S}_t | \frac{\|d - 1NN_{\mathcal{C}}(d)\|}{\|d - 2NN_{\mathcal{C}}(d)\|} < \lambda_{\mathcal{C}}\}$ ;
5   // Transitive matching with context
    $\mathcal{F}_t = \mathcal{T}_t^* \setminus \mathcal{C}_t^*$ ;
6    $(\hat{x}_t, \hat{y}_t, \hat{\theta}_t, \hat{s}_t) = \operatorname{argmin}_{x, y, \theta, s} \{ \sum_{f \in \mathcal{F}_t} L(e_f; M) \}$ ;

```

```

8   if ( $|\hat{s}_t - \hat{s}_{t-1}| < k_s$ ) and ( $|\hat{\theta}_t - \hat{\theta}_{t-1}| < k_\theta$ ) then
9     // object detected
10     $\hat{x}_t = (\hat{x}_t, \hat{y}_t, \hat{\theta}_t, \hat{s}_t)$ ;
11     $\mathcal{E}_t = \{(p, d) \in \mathcal{S}_t | p \in \text{OBB}(\hat{x}_t)\}$ ;
12     $\mathcal{O}_t = \{(p, d) \in \mathcal{C}_t^* | p \in \text{OBB}(\hat{x}_t)\}$ ;
13    // Occlusion detection
14    if  $|\mathcal{O}_t| \leq N_{\mathcal{O}}$  then
15      // Object non-occluded
16      // Object appearance update
17       $\mathcal{E}'_t = \{(p', d) | (p, d) \in \mathcal{E}_t, p' = M(p; \hat{x}_t)\}$ ;
18       $\mathcal{T}_t = \mathcal{T}_{t-1} \cup \mathcal{E}'_t$ ;
19      // Context appearance update
20       $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \mathcal{O}_t$ ;
21       $\mathcal{C}_t = \left( \bigcup_{\tau=t-l}^t (\mathcal{S}_\tau \setminus \mathcal{E}_\tau) \right) \cup \mathcal{D}_t$ ;
22      // Uniform random sampling forgetting features
23      if  $|\mathcal{T}_t| > N_{\mathcal{T}}$  then
24        | RandSamp( $\mathcal{T}_t, N_{\mathcal{T}}$ );
25      end
26      if  $|\mathcal{D}_t| > N_{\mathcal{D}}$  then
27        | RandSamp( $\mathcal{D}_t, N_{\mathcal{D}}$ );
28      end
29    end
30 until True;

```

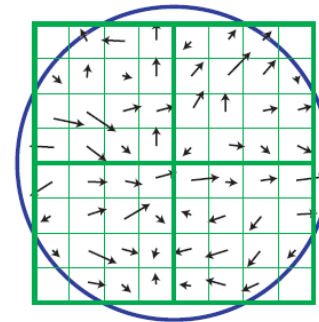
[Pernici, 2013] Pernici, Federico, and Alberto Del Bimbo. "Object tracking by oversampling local features." *IEEE transactions on pattern analysis and machine intelligence* 36.12 (2013): 2538-2551.

EXTRAER CARACTERÍSTICAS

2

Crop out the search region

$S_t = \{x \in \mathbb{R}^d : \|x - \hat{x}_{t-1}\|_\infty < r\}$ and extract local features $\mathcal{S}_t = \{(p, d) | p \in S_t, d \in \mathbb{R}^d\}$.



[Lowe, 2004]

- ALIEN usa el método SIFT para la representación local (extracción de descriptores)



Estado inicial

```
box = cv::Rect(x0, y0, w0, h0);
drawBox(&frame, box, CV_RGB(0, 255, 0), 1);
```



Características en el área de búsqueda



Características en el objeto a seguir

[Lowe, 2004] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.

SEGUIMIENTO DE OBJETOS BASADO EN TEMPLATES

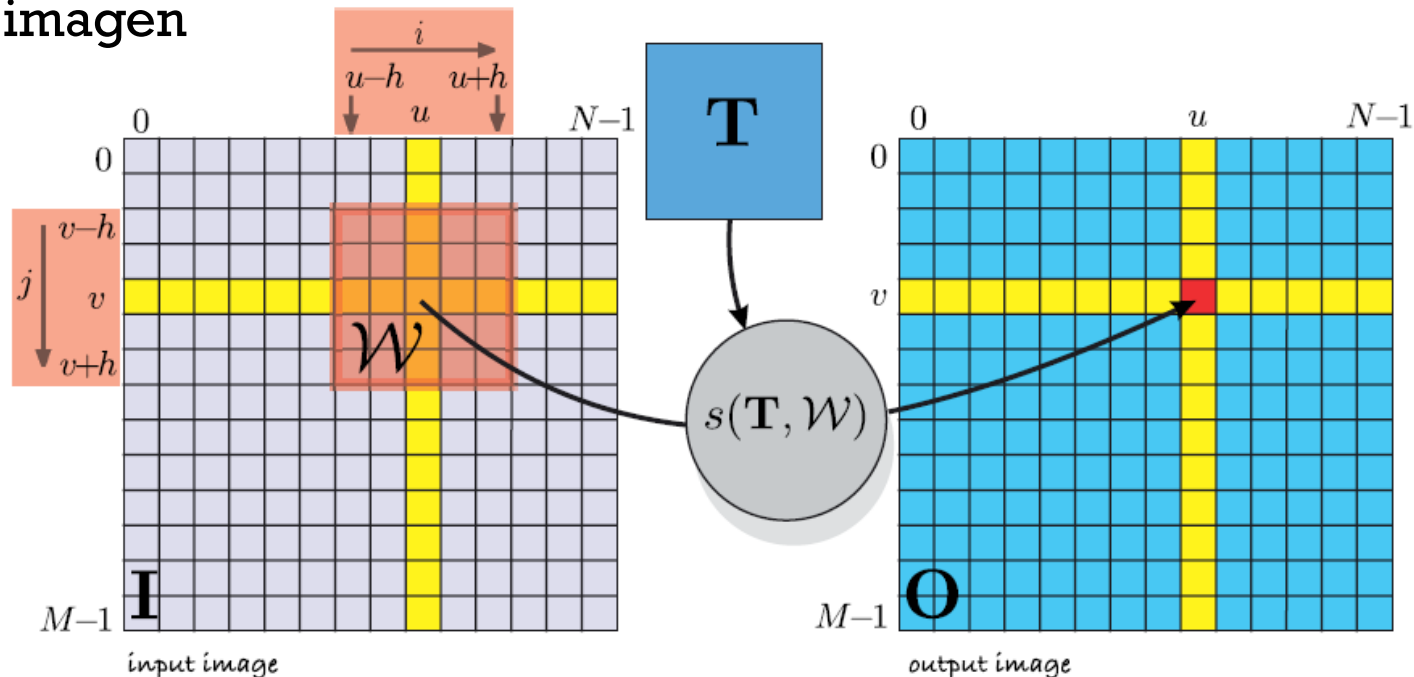
- Está relacionado con el seguimiento basado en regiones, ya que un *template* es esencialmente un modelo de la región de una imagen (objeto correspondiente) a seguir
- Inicialización
 - Seleccionar el *template* que deseamos seguir
 - Dicho *template* puede ser generado de diversas maneras:
 - a) Por ejemplo, en una aplicación de video llamadas, es posible adquirir una foto del usuario en tiempo real, entonces el sistema detecta el rostro y esta región puede usarse como un *template*
 - b) También puede ser generado offline, empleando métodos estadísticos, por ejemplo, incorporando información de diversas bases de datos de rostros y entonces usar el promedio o los eigenrostros como *template*

SEGUIMIENTO DE OBJETOS BASADO EN TEMPLATES (C1)

- El emparejamiento del *template* puede ser definido como el proceso de buscar la región de la imagen que mejor se parece al *template* con base en una medida de distancia o similitud
- El objetivo es encontrar los parámetros de transformación que hacen que el *template* se ajuste a cierta región de la imagen
- Para evitar una búsqueda exhaustiva podemos usar:
 - **Substracción de fondo.** Para determinar las regiones de la imagen en donde hay cierta actividad (movimiento, cambios, etc.)
 - **Predecir el movimiento.** Por ejemplo, usando el filtro de Kalman, podemos predecir hacia a donde va nuestro objeto en el siguiente *frame* del video

SEGUIMIENTO DE OBJETOS BASADO EN TEMPLATES (C2)

- Sea T_i la intensidad del i -ésimo pixel (x_i, y_i) en el template
- Sea $I_{i,W}$ la intensidad del i -ésimo pixel (x_i, y_i) en la región W de la imagen



SEGUIMIENTO DE OBJETOS BASADO EN TEMPLATES (C3)

- Sea M el número de píxeles en el *template*
- Podemos calcular el “*matching*” buscando la región v de la imagen que minimiza alguna de las siguientes métricas:

Suma de diferencias absolutas (SAD)

$$\sum_{i=1}^M |I_{i,v} - T_i|$$

Suma de diferencias al cuadrado (SSD)

$$\sum_{i=1}^M (I_{i,v} - T_i)^2$$

Entropía conjunta

$$H(I_v, T) = - \sum_{i=1}^{nb} \sum_{j=1}^{nb} h(i, j) \log(h(i, j))$$

donde h es el histograma normalizado conjunto entre I_v y T , con nb el número de bins

- O que maximiza:

Correlación cruzada normalizada

$$\frac{\sum_{i=1}^M (T_i - \bar{T})(I_{i,v} - \bar{I}_v)}{\sqrt{\sum_{i=1}^M (T_i - \bar{T})^2} \sqrt{\sum_{i=1}^M (I_{i,v} - \bar{I}_v)^2}}$$

con \bar{I}_v y \bar{T} los promedios de sus respectivas regiones

Alan C. Bovik. 2009. *The Essential Guide to Video Processing* (2nd ed.). Academic Press

Wilhelm Burger, Mark J. Burge. 2009. *Principles of Digital Image Processing Core Algorithms*. Springer-Verlag London.

Seguimiento de objetos (Met. Tradicionales). Francisco J. Hernández-López

Ago-Dic 2025

EJEMPLOS

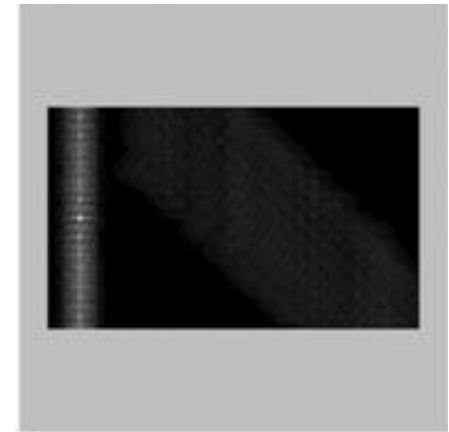
Input image



Template



Correlation result

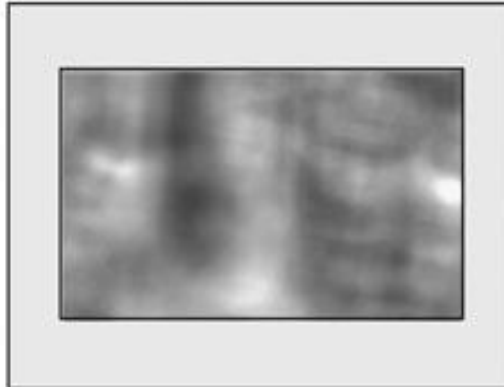


Template processing

Correlation



Normalized cross correlation



Input image



Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Seguimiento de objetos (Met. Tradicionales). Francisco J. Hernández-López

Ago-Dic 2025

¿DÓNDE ESTÁ WALLE?



eva



dynomutt



bender



DEFINIENDO EL OBJETO A SEGUIR Y EL ÁREA DE BÚSQUEDA

- El rectángulo verde es la caja envolvente (Bounding Box o BBox), que encierra al objeto que queremos seguir a través del video

- $BBox = (x, y, W, H)$

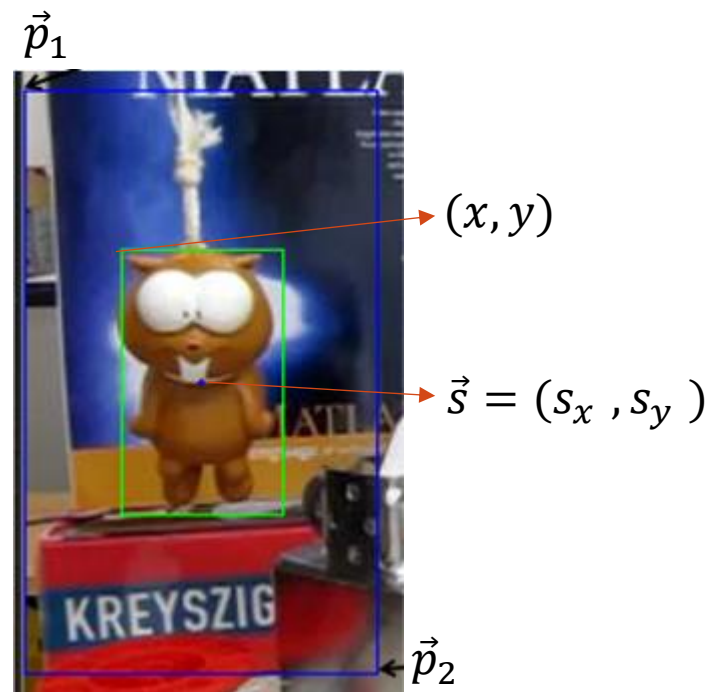
- $s_x = x + \frac{W}{2}, s_y = y + \frac{H}{2}$

- Entonces, definimos $T = I^1(BBox)$

- El objetivo es buscar el nuevo estado \vec{s}^t al tiempo t , en una región de la imagen centrada en el estado \vec{s}^{t-1}

- $\vec{p}_1^t = (s_x^{t-1} - \rho W, s_y^{t-1} - \rho H),$

- $\vec{p}_2^t = (s_x^{t-1} + \rho W, s_y^{t-1} + \rho H)$



ALGORITMO DE EMPAREJAMIENTO DE BLOQUES

Para $t = 2$ hasta $t \leq N$

$$\vec{p}_1^t = (s_x^{t-1} - \rho W, s_y^{t-1} - \rho H), \vec{p}_2^t = (s_x^{t-1} + \rho W, s_y^{t-1} + \rho H)$$

$$ssd_{min} = 255 \times 255 \times W \times H$$

Para $x = p_{1x}$ hasta $x \leq p_{2x} - W$

Para $y = p_{1y}$ hasta $y \leq p_{2y} - H$

$$I_{sub}^t = I^t(x, y, W, H)$$

$$ssd = \sum_i^W \sum_j^H \left(I_{sub}^t(i, j) - T(i, j) \right)^2$$

Si $(ssd < ssd_{min})$ entonces

$$ssd_{min} = ssd$$

$$s_x^t = x + \frac{W}{2}, s_y^t = y + \frac{H}{2}$$

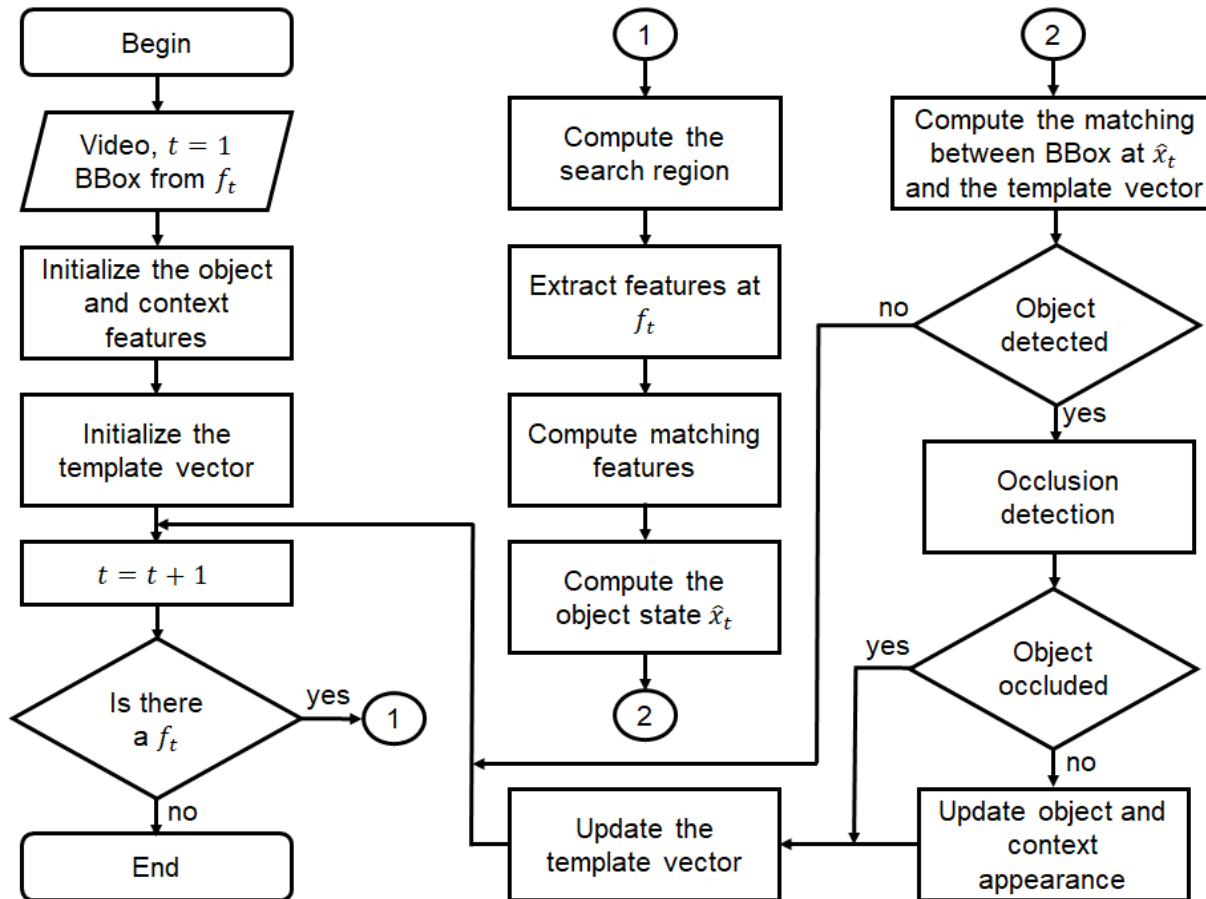
Fin Si

Fin Para

Fin Para

Fin Para

SEG. DE OBJ. USANDO PUNTOS CARACT. Y EMPAREJ. DE BLOQUES (FPDTM)

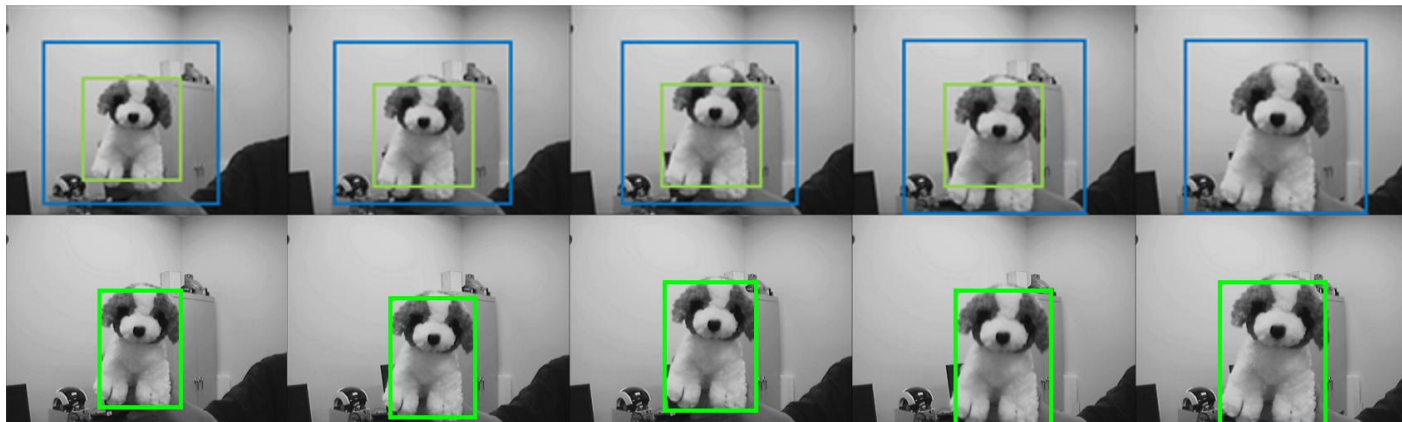


Pat-Chan, A. E., Hernandez-Lopez, F. J., & Moreno-Sabido, M. R. (2025). Video Object Tracking by Feature Point Descriptor and Template Matching. *Computación y Sistemas*, 29(3).

Seguimiento de objetos (Met. Tradicionales). Francisco J. Hernández-López

Ago-Dic 2025

COMPARACIÓN CUALITATIVA

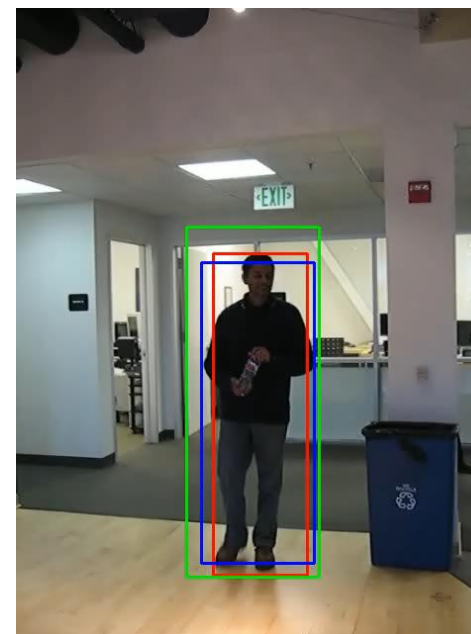


TM

FPDTM



FPDTM



Rojo: GT
Verde: ROLO
Azul: FPDTM

COMPARACIÓN CUANTITATIVA

Table 2. Results of the recall metric using the TM, FPDTM, and YOLOV5-StrongSORT methods with different values of λ : $\lambda_l|\lambda_m|\lambda_h$

ID	Sequence	TM	FPDTM	YOLOV5-StrongSORT
1	Ball	0.09 0.09 0.03	0.40 0.34 0.19	0.31 0.31 0.31
2	Board	0.05 0.04 0.03	0.14 0.13 0.07	NA NA NA
3	Box	0.14 0.14 0.03	0.78 0.73 0.45	NA NA NA
4	Car	0.18 0.09 0.03	0.46 0.20 0.05	0.99 0.99 0.73
5	Car_2	0.29 0.04 0.03	0.98 0.98 0.93	0.57 0.55 0.01
6	Carchase	0.04 0.04 0.02	0.17 0.16 0.09	0.01 0.0 0.0
7	Cup_on_table	0.13 0.13 0.07	0.93 0.71 0.47	0.40 0.40 0.40
8	Dog1	0.11 0.11 0.09	0.67 0.62 0.49	0.98 0.97 0.51
9	Gym	0.18 0.15 0.03	0.60 0.27 0.04	0.94 0.89 0.48
10	Juice	1.00 0.47 0.32	1.00 1.00 1.00	NA NA NA
11	Jumping	0.99 0.71 0.17	0.81 0.71 0.14	NA NA NA
12	Lemming	0.04 0.03 0.03	0.72 0.31 0.20	NA NA NA
13	Liquor	0.17 0.17 0.17	0.86 0.86 0.84	1.00 1.00 0.90
14	Mountain-bike	0.26 0.26 0.12	0.49 0.21 0.11	0.08 0.07 0.00
15	Person	0.10 0.10 0.05	0.72 0.64 0.35	1.00 1.00 1.00
16	Person_crossing	0.06 0.06 0.06	0.86 0.84 0.75	0.64 0.63 0.59
17	Person_p_occluded	0.53 0.53 0.53	1.00 0.95 0.82	0.99 0.99 0.99
18	Singer	0.17 0.17 0.12	0.34 0.34 0.27	0.23 0.19 0.00
19	Sylvester	0.05 0.05 0.03	0.51 0.45 0.35	NA NA NA
20	Track_running	0.28 0.24 0.11	0.16 0.10 0.06	0.93 0.93 0.57

GRACIAS POR SU ATENCIÓN

Francisco J. Hernández López

fcoj23@ciimat.mx

WebPage:

www.ciimat.mx/~fcoj23

