



Ciencia y Tecnología

Secretaría de Ciencia, Humanidades, Tecnología e Innovación



CIMAT
UNIDAD MÉRIDA

SEGUIMIENTO DE OBJETOS EN UN VIDEO

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

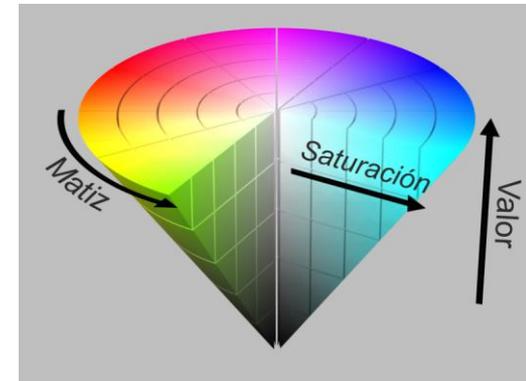
SEGUIMIENTO DE OBJETOS RÍGIDOS 2D

- 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

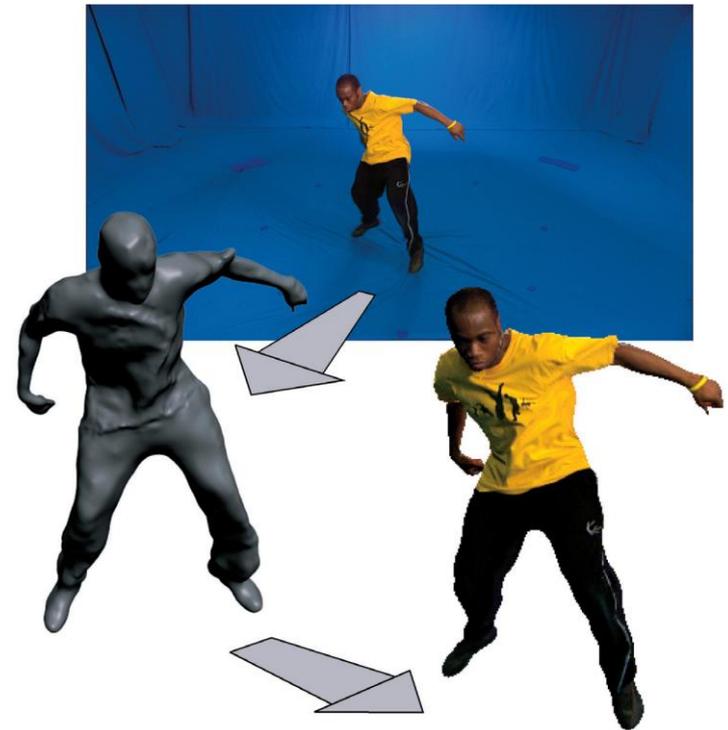
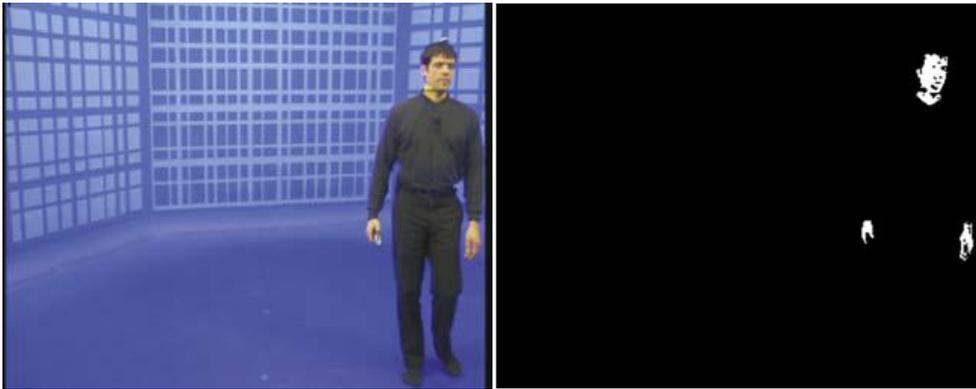
$$\frac{R}{R + G + B}, \frac{G}{R + G + B}, \frac{B}{R + G + B}$$



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

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

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

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

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

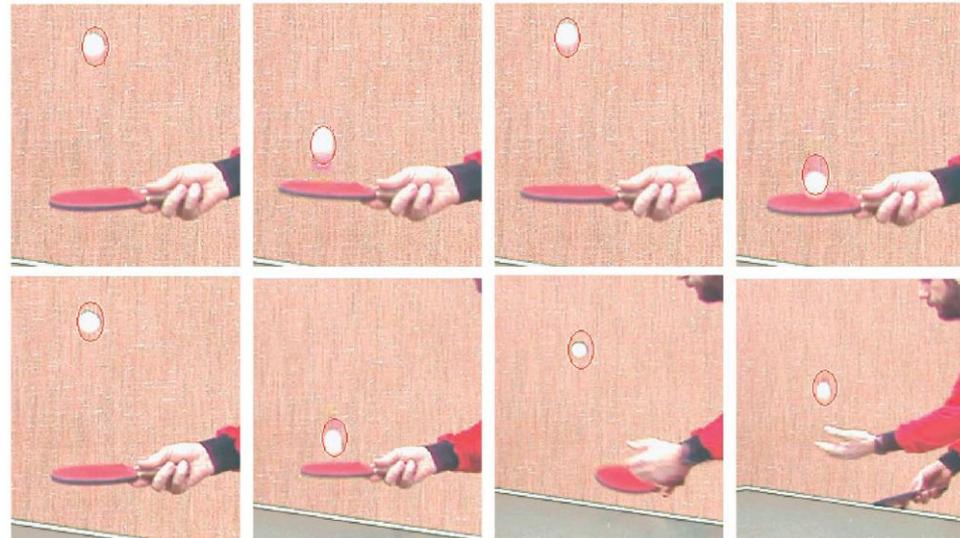
- a) Actualizar $\vec{y}_1 \leftarrow \frac{1}{2}(\vec{y}_0 + \vec{y}_1)$
b) Evaluar $\rho[\hat{p}(\vec{y}_1), \hat{q}]$

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

- a) parar

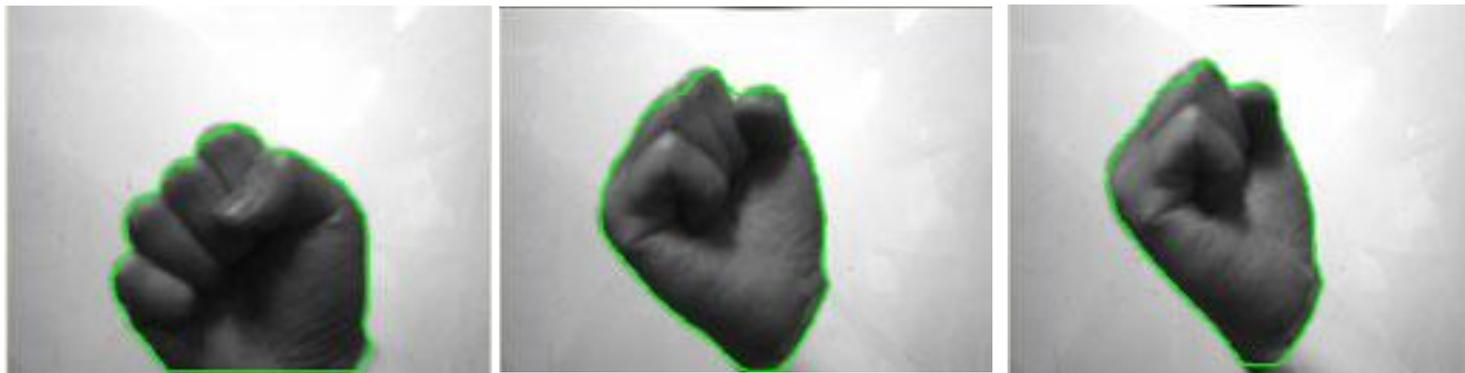
De lo contrario

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



SEGUIMIENTO DE OBJETOS BASADO EN CONTORNOS

- Una alternativa para el seguimiento de los objetos es utilizar la información de su contorno e intentar seguir el contorno a través del tiempo: recupera tanto la posición como la forma
- Es más complicado que modelar regiones enteras, sin embargo, es robusto a oclusiones y cambios de iluminación en la escena
- Usado tanto para objetos rígidos como deformables



Lee, Jung-Ho, Fang Hua, and Jong Whan Jang. "An Improved Object Detection and Contour Tracking Algorithm Based on Local Curvature." *Signal Processing, Image Processing and Pattern Recognition*. Springer Berlin Heidelberg, 2009. 25-32.

Seguimiento de objetos. Francisco J. Hernández-López

Ene-May 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

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

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

SEGUIMIENTO DE OBJETOS BASADO EN TEMPLATES

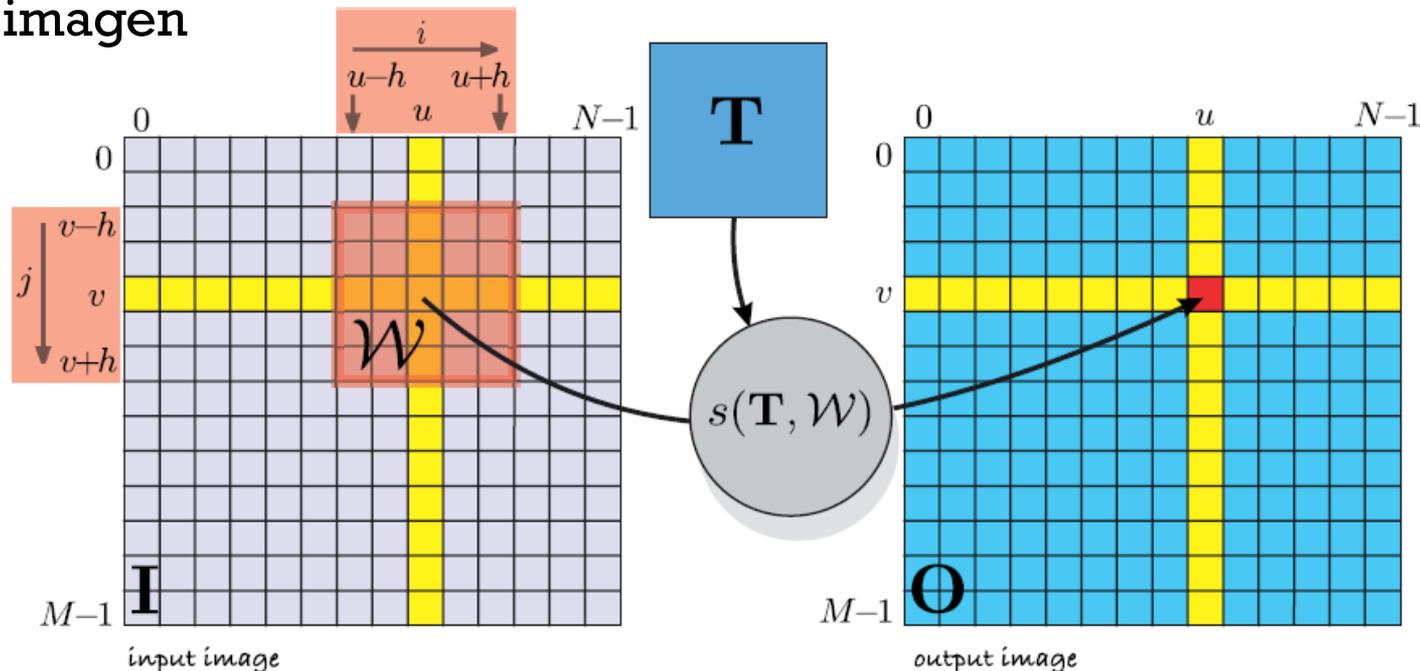
- Está relacionado con el seguimiento basado en regiones, ya que un *template* es esencialmente un modelo de la región de una imagen (objeto correspondiente) a seguir
- Inicialización
 - Seleccionar el *template* que deseamos seguir
 - Dicho *template* puede ser generado de diversas maneras:
 - a) Por 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

EJEMPLOS

Input image



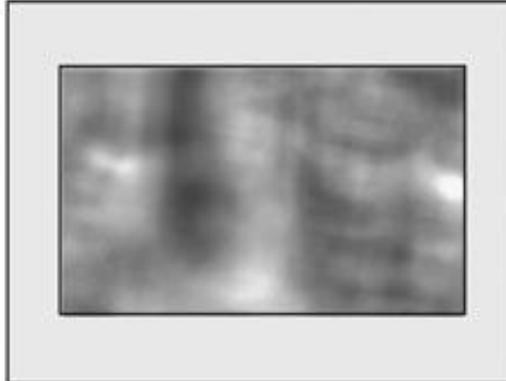
Template



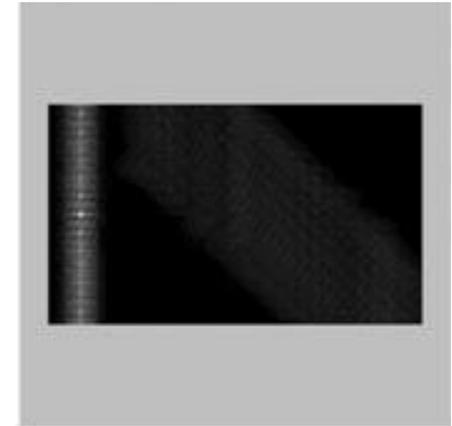
Correlation



Normalized cross correlation

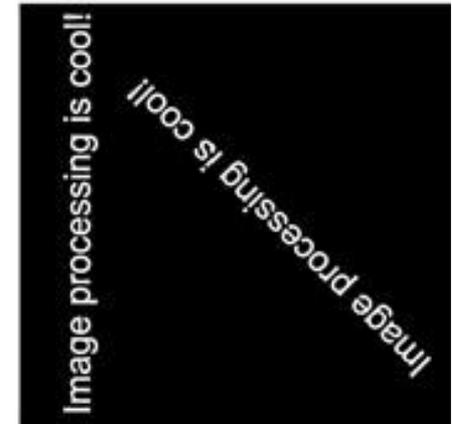


Correlation result



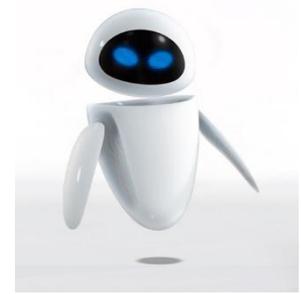
Template
processing

Input image



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

¿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

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

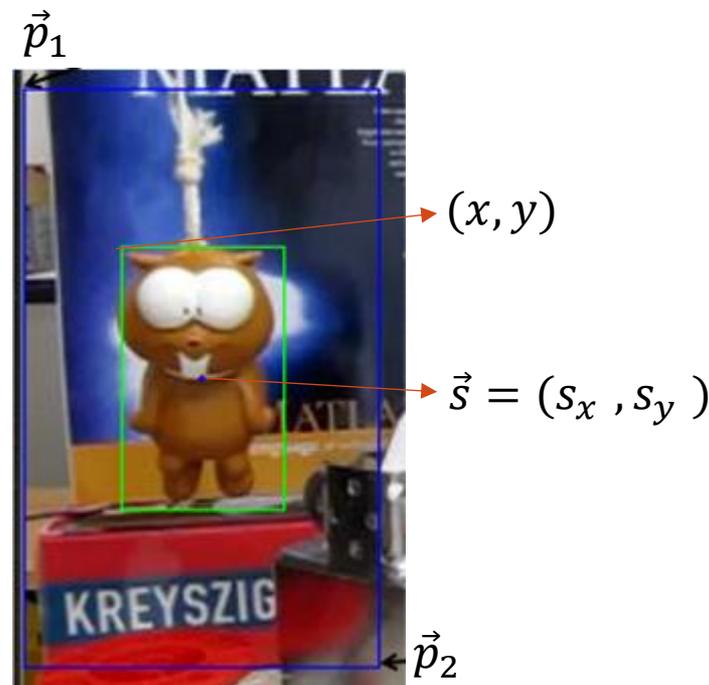
- $s_x = x + \frac{W}{2}, s_y = y + \frac{H}{2}$

- 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

GRACIAS POR SU ATENCIÓN

Francisco J. Hernández López

fcoj23@ciimat.mx

WebPage:

www.ciimat.mx/~fcoj23

