

ALGORITMOS DE ESTIMACIÓN DE MOVIMIENTO

Francisco J. Hernández López

fcoj23@cimat.mx



ESTIMACIÓN DE MOVIMIENTO GLOBAL

- El movimiento de la cámara afecta el movimiento de todos los píxeles de la secuencia de video
- Un algoritmo de compensación de movimiento global (CMG) puede ser incluido en algún compresor de video (como MPEG)
- Bajo el modelo de intensidad constante y el criterio de error cuadrático, tenemos el siguiente problema de optimización:

$$\min_{\vec{b}} \mathcal{E}(\vec{v}), \quad \mathcal{E}(\vec{v}) = \sum_{\vec{x}} \mathcal{E}^2(\vec{x}), \quad \mathcal{E}(\vec{x}) = I_k(\vec{x}) - I_{k-1}[\vec{x} - \vec{v}(\vec{x})(t_k - t_{k-1})]$$

asumiendo $t_k - t_{k-1} = 1$

Algunas elecciones de $\vec{v}(\vec{x})$ pueden ser:

$$\begin{cases} \vec{v}(\vec{x}) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \\ \vec{v}(\vec{x}) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} b_3 & b_4 \\ b_5 & b_6 \end{pmatrix} \vec{x} \end{cases}$$

ESTIMACIÓN DE MOVIMIENTO GLOBAL (C1)

- Ya que la dependencia entre $\mathcal{E}(\vec{v})$ y \vec{b} es no lineal, entonces se utiliza un procedimiento de minimización iterativo, por ejemplo:

$$\vec{b}^{k+1} = \vec{b}^k + B^{-1}\vec{c}$$

donde B es la matriz Hessiana de $K \times K$

$$B_{k_1 k_2} = \sum_{\vec{x}} \frac{\partial^2 \mathcal{E}^2(\vec{x})}{\partial b_{k_1} \partial b_{k_2}}$$

y \vec{c} un vector

$$\vec{c} = - \sum_{\vec{x}} \mathcal{E}(\vec{x}) \frac{\partial \mathcal{E}(\vec{x})}{\partial b_{k_1}}$$

EMPAREJAMIENTO DE BLOQUES

- Es el algoritmo más simple para estimación de movimiento local
- Utiliza un modelo de movimiento constante en espacio y lineal en el tiempo sobre una región de soporte rectangular
- Puede ser implementado en tiempo real usando VLSI (*Very Large Scale Integration*), CUDA (*Compute Unified Device Architecture*), etc.
- El objetivo es minimizar:

$$\min_{\vec{d}(\vec{x}) \in S} \mathcal{E}(\vec{d}(\vec{x})), \quad \mathcal{E}(\vec{d}(\vec{x})) = \sum_{\vec{y} \in R_{\vec{x}}} \Phi[I_k(\vec{y}) - I_{k-1}(\vec{y} - d(\vec{x}))], \quad \forall \vec{x}$$

donde S es el área de búsqueda en la cual $d(\vec{x}) \in S$

CORRELACIÓN DE FASE, PHASE ONLY CORRELATION (POC)

- Método basado en el dominio de la frecuencia
- Se utiliza para estimar el movimiento global
- En combinación con el emparejamiento de bloques, POC explota las ventajas de los criterios tanto en el dominio espacial como en la frecuencia
- La idea es calcular los candidatos \vec{d} en el dominio de la frecuencia y entonces realizar una búsqueda con emparejamiento de bloques solo para dichos candidatos (evitando la búsqueda exhaustiva)

POC (C1)

- Dado el criterio de correlación cruzada

$$C(\vec{d}) = \sum_{\vec{x}} f(\vec{x})g(\vec{x} - \vec{d}(\vec{x})),$$

- Usando Fourier puede ser expresado como

$$\mathcal{F}(C(\vec{d})) = \mathcal{F}\left[\sum_{\vec{x}} f(\vec{x})g(\vec{x} - \vec{d}(\vec{x}))\right] = \mathcal{F}(\omega)G^*(\omega),$$

donde $G^*(\omega)$ es el conjugado de $G(\omega)$

- Normalizando $\mathcal{F}(C(\vec{d}))$ y tomando la inversa de Fourier

$$r(\vec{x}) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}(\omega)G^*(\omega)}{|\mathcal{F}(\omega)G^*(\omega)|}\right\}$$

Entonces $r(\vec{x})$ es la correlación normalizada entre f y g

POC (C2)

- En el caso especial de una traslación global

$$f(\vec{x}) = g(\vec{x} - \vec{d}),$$

- Usando la propiedad del corrimiento podemos que ver que la inversa de la transformada de Fourier de una exponencial compleja es una función delta de Kronecker:

$$r(\vec{x}) \Big|_{f(\vec{x})=g(\vec{x}-\vec{d})} = \delta(\vec{x} - \vec{d})$$

- En la práctica $r(\vec{x})$ no es solo un impulso, tiene numerosos picos, los cuales pueden ser utilizados como candidatos en un emparejamiento por bloques

ALGORITMO POC CON EMPAREJAMIENTO DE BLOQUES

1. Dividir I_{k-1} y I_k en bloques grandes I_{k-1}^m y I_k^m (64×64 considerando movimientos de hasta ± 32 píxeles) para $m = 1, \dots, M$, con M el número de bloques
2. Calcular la transformada de Fourier de cada bloque
3. Calcular $r^m(\vec{x})$ con los correspondientes bloques I_{k-1}^m y I_k^m
4. Estimar n desplazamientos candidatos \vec{d}_n^m en cada bloque, a partir de los picos más altos de $r^m(\vec{x})$, (por ej. con $n = 5$)
5. Utilizar la estrategia de emparejamiento de bloques con los \vec{d}_n^m candidatos para obtener el \vec{d}^* óptimo en cada pixel (por ej. Con ventanas de 16×16)

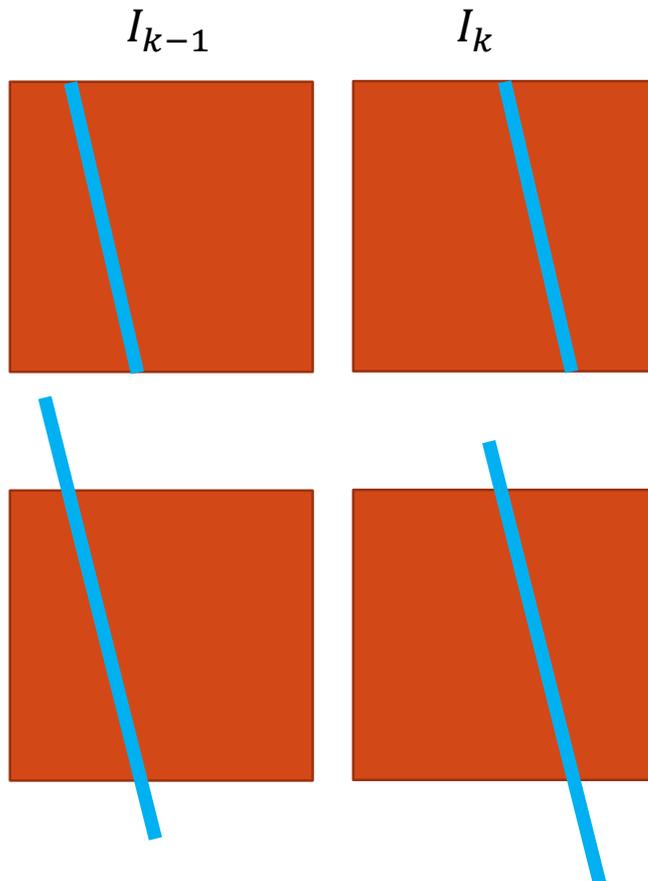
FLUJO ÓPTICO



<http://vision.middlebury.edu/flow/data/>

- Patrón de movimiento aparente de los objetos, superficies y bordes en una escena, causado por el movimiento relativo entre un observador (un ojo o cámara) y la escena
https://es.wikipedia.org/wiki/Flujo_óptico
- Refleja los cambios en la imagen debido al movimiento en un intervalo de tiempo dt , el campo del flujo óptico es el campo de velocidades que representan el movimiento 3D de los objetos a través de una imagen 2D
- Suposiciones:
 - El brillo observado de un objeto es constante en el tiempo
 - Puntos cercanos en el plano de la imagen se mueven de forma similar (restricción de suavidad en las velocidades)

PROBLEMA DE LA APERTURA



Movimiento percibido
Solo puedo ver en
dirección del gradiente

Movimiento real

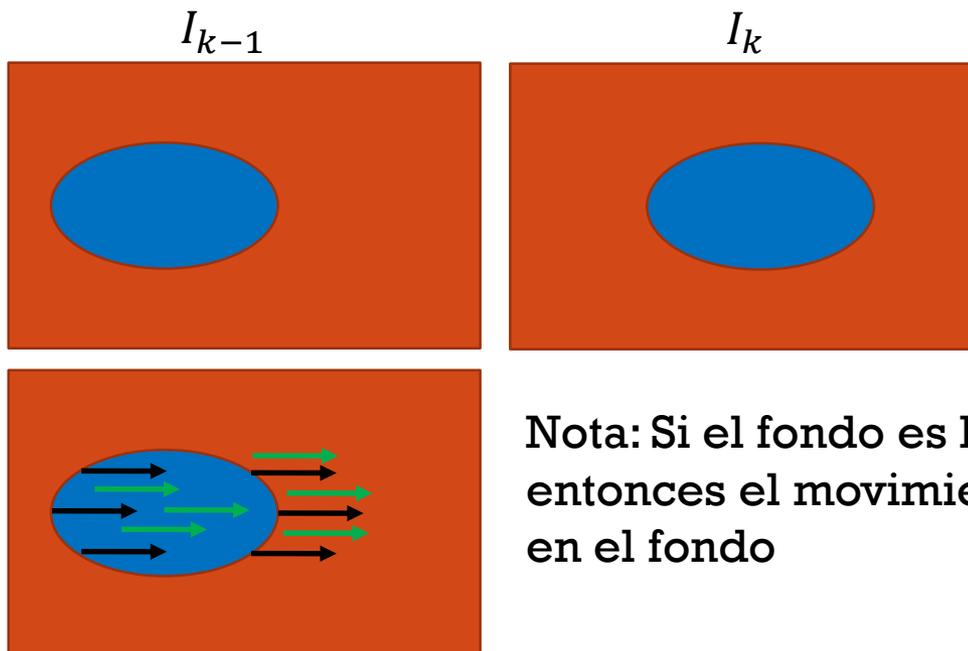


barber's pole
https://en.wikipedia.org/wiki/Barberpole_illusion

REGULARIZACIÓN

- Por el problema de la apertura, podemos agregar un término de regularización

$$\vec{d}(\vec{x}) \approx \vec{d}(\vec{y}), \text{ con } \|\vec{x} - \vec{y}\| = 1$$



Nota: Si el fondo es liso (no tiene textura), entonces el movimiento se va a chorrear en el fondo

FLUJO ÓPTICO (HORN AND SCHUNK, 1981)

- Minimizar el siguiente error:

$$E^2(x_1, x_2) = \underbrace{\left(\frac{\partial f}{\partial x_1} v_1 + \frac{\partial f}{\partial x_2} v_2 + \frac{\partial f}{\partial t} \right)^2}_{\text{Término de datos}} + \lambda \underbrace{\left(\left(\frac{\partial v_1}{\partial x_1} \right)^2 + \left(\frac{\partial v_1}{\partial x_2} \right)^2 + \left(\frac{\partial v_2}{\partial x_1} \right)^2 + \left(\frac{\partial v_2}{\partial x_2} \right)^2 \right)}_{\text{Término de restricción de suavidad}}$$

- Esto se reduce a resolver las siguientes ecuaciones:

$$\left(\lambda^2 + \left(\frac{\partial f}{\partial x_1} \right)^2 \right) v_1 + \frac{\partial f}{\partial x_1} \frac{\partial f}{\partial x_2} v_2 = \lambda^2 \bar{v}_1 - \frac{\partial f}{\partial x_1} \frac{\partial f}{\partial t},$$

$$\frac{\partial f}{\partial x_1} \frac{\partial f}{\partial x_2} v_1 + \left(\lambda^2 + \left(\frac{\partial f}{\partial x_2} \right)^2 \right) v_2 = \lambda^2 \bar{v}_2 - \frac{\partial f}{\partial x_2} \frac{\partial f}{\partial t},$$

Donde \bar{v}_1 y \bar{v}_2 son los valores promedio de las velocidades en las direcciones x_1 y x_2 en alguna vecindad de (x_1, x_2)

ALGORITMO DE FLUJO ÓPTICO (HS)

1. Inicializar velocidades $v_1(i, j) = 0$ y $v_2(i, j) = 0$ para $(i, j) \in \Omega$
2. Dado el número de iteraciones k , calcular los valores de v_1^k y v_2^k para todos los pixeles (i, j)

$$v_1^k(i, j) = \bar{v}_1^{k-1}(i, j) - \frac{\partial f(i, j)}{\partial x_1} \frac{P(i, j)}{D(i, j)},$$

$$v_2^k(i, j) = \bar{v}_2^{k-1}(i, j) - \frac{\partial f(i, j)}{\partial x_2} \frac{P(i, j)}{D(i, j)},$$

donde $P(i, j) = \frac{\partial f(i, j)}{\partial x_1} \bar{v}_1 + \frac{\partial f(i, j)}{\partial x_2} \bar{v}_2 + \frac{\partial f(i, j)}{\partial t}$, $D(i, j) = \lambda^2 + \left(\frac{\partial f(i, j)}{\partial x_1}\right)^2 + \left(\frac{\partial f(i, j)}{\partial x_2}\right)^2$

3. Parar si

$$\sum_i \sum_j E^2(i, j) < \epsilon,$$

donde ϵ es el mínimo error permitido, en otro caso ir al paso 2

FLUJO ÓPTICO (LUCAS-KANADE, 1981)

- Asumiendo intensidad constante

$$I_k(x_1 + d_1, x_2 + d_2) - I_{k-1}(x_1, x_2) = 0$$

- Asumiendo que los desplazamientos son pequeños entonces

$$I_k(x_1 + d_1, x_2 + d_2) \approx I_k(x_1, x_2) + \frac{\partial I_k}{\partial x_1} d_1 + \frac{\partial I_k}{\partial x_2} d_2$$

$$I_k(x_1, x_2) - I_{k-1}(x_1, x_2) + \frac{\partial I_k}{\partial x_1} d_1 + \frac{\partial I_k}{\partial x_2} d_2 = 0$$

- Asumiendo que en la vecindad de un pixel (x_1, x_2) los desplazamientos son similares:
 - Por ejemplo, en una ventana de 3×3 centrada en (x_1, x_2) el movimiento es (d_1, d_2) , entonces habrían 9 ecuaciones lineales para calcular los desplazamientos

FLUJO ÓPTICO (LUCAS-KANADE, 1981) (C1)

- Podemos construir

$$A \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \vec{c}$$

con A una matriz de 9×2 , donde cada fila es una estimación de

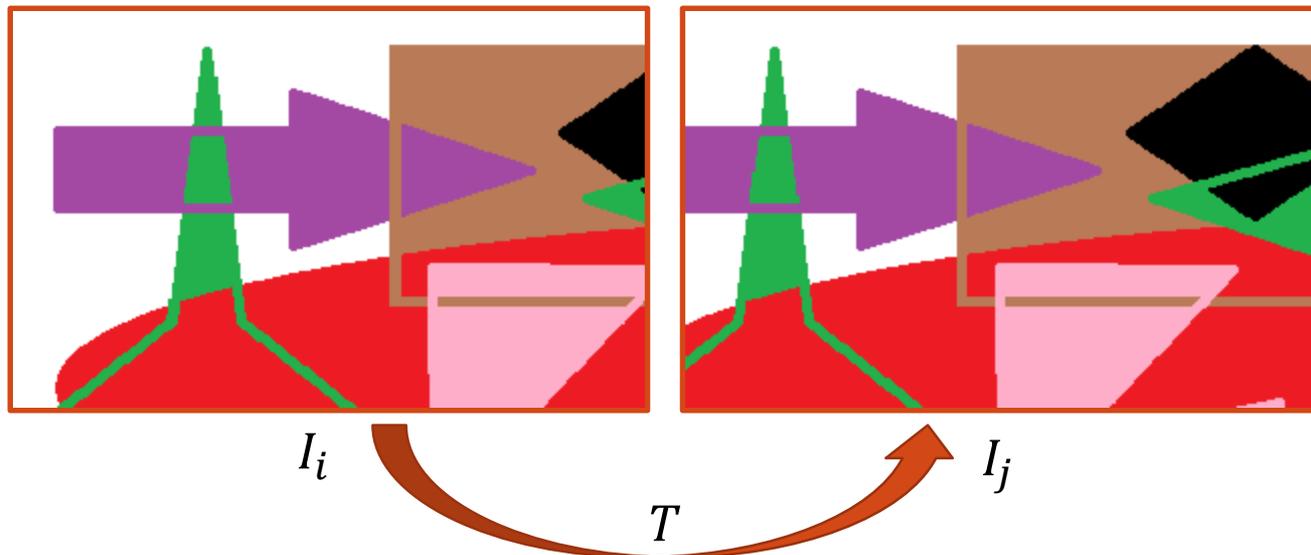
$\frac{\partial I_k}{\partial x_1}, \frac{\partial I_k}{\partial x_2}$ y \vec{c} es un vector de 9×1 en donde cada componente es la diferencia de intensidades entre I_k y I_{k-1} en los respectivos pixeles

- La solución de este sistema puede ser

$$A^T A \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = A^T \vec{c}, \quad A^T A = \begin{pmatrix} \sum \frac{\partial I_k}{\partial x_1} \frac{\partial I_k}{\partial x_1} & \sum \frac{\partial I_k}{\partial x_1} \frac{\partial I_k}{\partial x_2} \\ \sum \frac{\partial I_k}{\partial x_2} \frac{\partial I_k}{\partial x_1} & \sum \frac{\partial I_k}{\partial x_2} \frac{\partial I_k}{\partial x_2} \end{pmatrix}$$

el cual es un sistema de 2×2 que tiene solución si $(A^T A)^{-1} \exists$

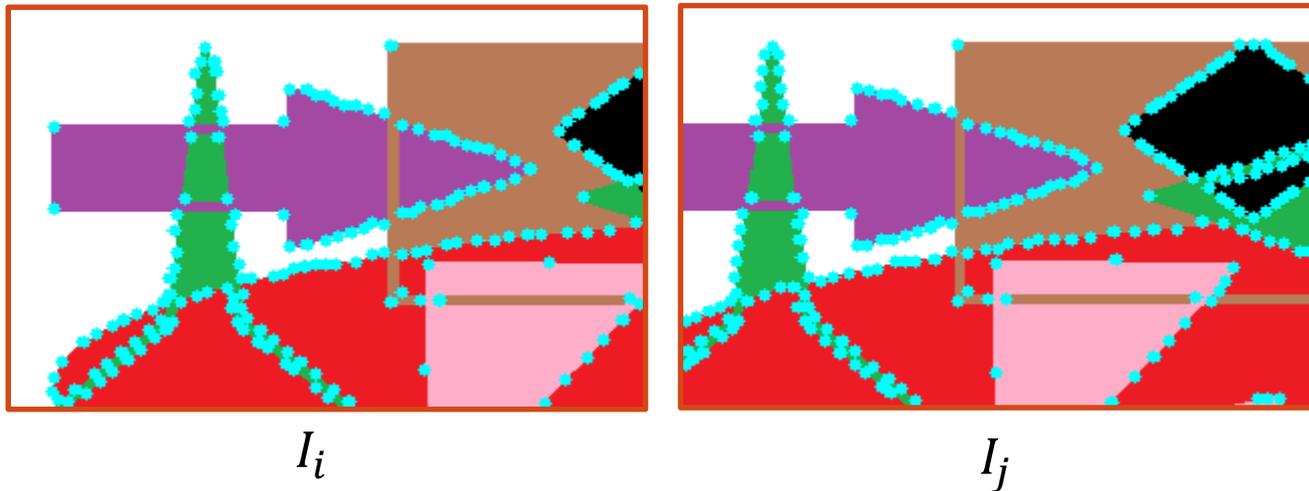
REGISTRO DE IMÁGENES



La idea es encontrar una transformación T , que relacione ambas imágenes, con el fin de poder traslaparlas. Existen diferentes métodos, estos se aplican o se construyen a partir de los diferentes niveles de procesamiento de la imagen:

- Por pixel
- Por regiones
- Por características, etc.

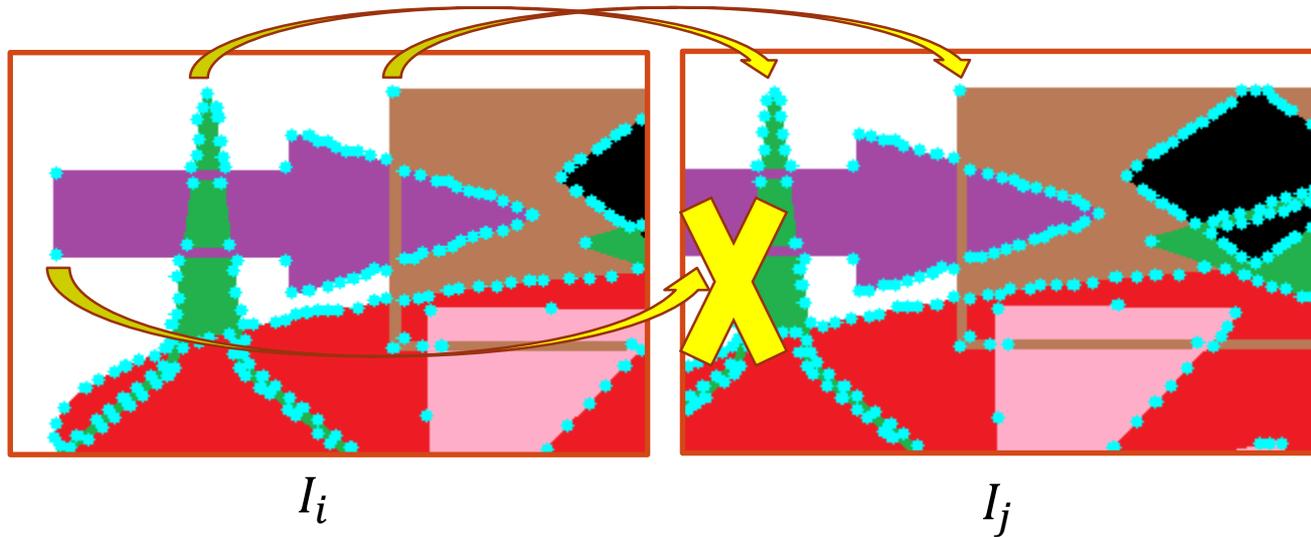
REGISTRO DE IMÁGENES POR CARACTERÍSTICAS



Detección de características en las imágenes

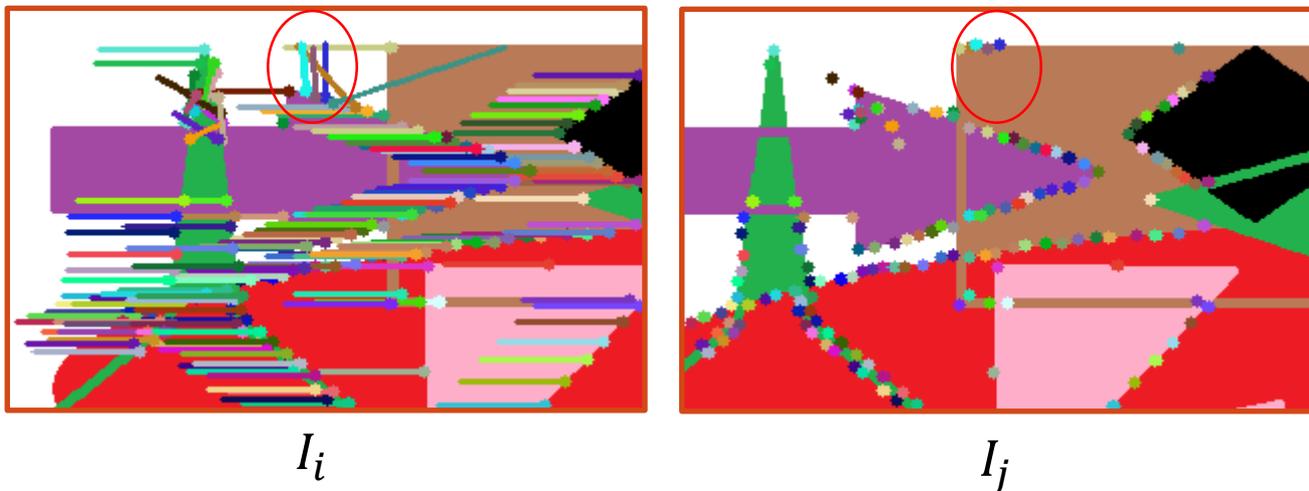
- `goodFeaturesToTrack()`
- `SIFT()`
- `SURF()`, etc.

REGISTRO DE IMÁGENES POR CARACTERÍSTICAS (C1)



Calculo de correspondencias entre las características

REGISTRO DE IMÁGENES POR CARACTERÍSTICAS (C2)



Calculo de correspondencias entre las características

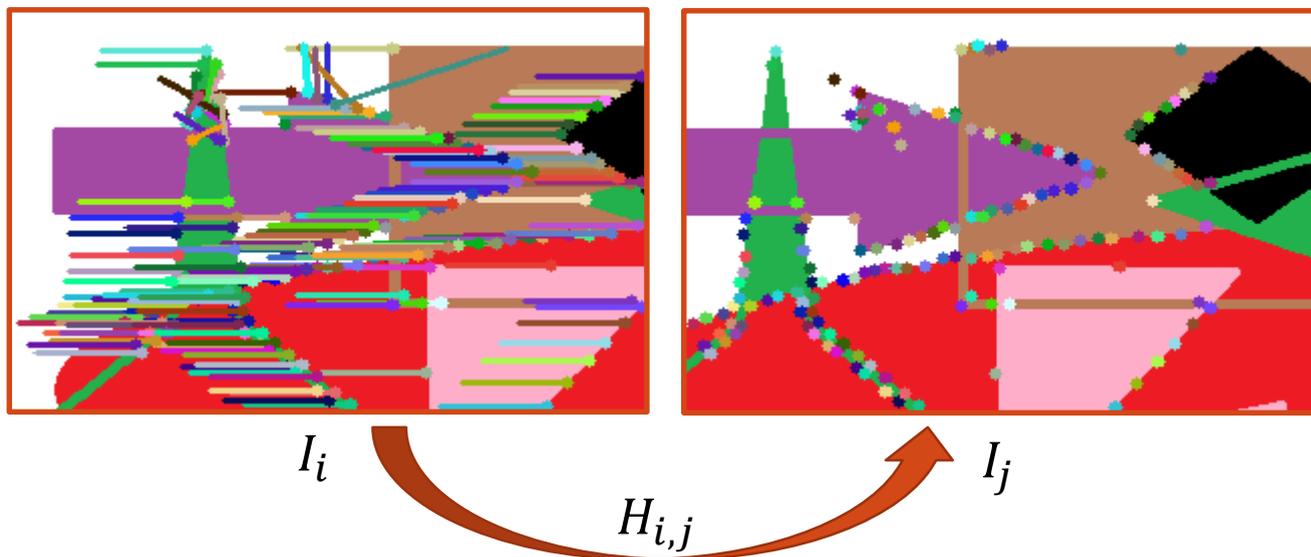
- Flujo óptico (Lucas-Kanade Piramidal)

Distancia Euclidiana entre los descriptores de las características:

$$d(c_r^i, c_s^j) = \|c_r^i - c_s^j\|_2$$

- Devolver todos los vectores con $d < threshold$
- Nearest Neighbor (NN): Vector con d más pequeña
- Nearest Neighbor Distance Ratio (NNDR): Si $NNDR < threshold$
 $NNDR = \frac{d_1}{d_2}$, con d_1 el más pequeño y d_2 el segundo más pequeño

REGISTRO DE IMÁGENES POR CARACTERÍSTICAS (C3)



Estimación de una Homografía $H_{i,j}$ a partir de las correspondencias

- $H_{i,j} \in \mathbb{R}^{3 \times 3}$
- Sea $\mathbf{x} = (x_1, x_2)^T$ la posición de un pixel en coordenadas cartesianas y $\hat{\mathbf{x}} = (\mathbf{x}^T, 1)^T$ su correspondiente punto en coordenadas homogéneas, entonces
$$\hat{\mathbf{x}}^j = H_{i,j} \hat{\mathbf{x}}^i$$

- Necesitamos por lo menos 4 correspondencias
- Podemos usar la técnica DLT (Direct Linear Transform)
- RANSAC (RANdom SAmple Consensus)

Hartley, A., Zisserman, A.: Multiple view geometry in computer vision (2 ed.). Cambridge University Press, Cambridge (2006)

CALCULAR LA HOMOGRAFÍA

- Tenemos que $\hat{x}^j = H_{i,j}\hat{x}^i$

$$\begin{pmatrix} x_1^j \\ x_2^j \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1^i \\ x_2^i \\ 1 \end{pmatrix}$$

- Entonces

$$x_1^j = \frac{h_{11}x_1^i + h_{12}x_2^i + h_{13}}{h_{31}x_1^i + h_{32}x_2^i + h_{33}}, \quad x_2^j = \frac{h_{21}x_1^i + h_{22}x_2^i + h_{23}}{h_{31}x_1^i + h_{32}x_2^i + h_{33}}$$

- Por cada correspondencia tenemos:

$$h_{11}x_1^i + h_{12}x_2^i + h_{13} - x_1^j(h_{31}x_1^i + h_{32}x_2^i + h_{33}) = 0$$

$$h_{21}x_1^i + h_{22}x_2^i + h_{23} - x_2^j(h_{31}x_1^i + h_{32}x_2^i + h_{33}) = 0$$

CALCULAR LA HOMOGRAFÍA (C1)

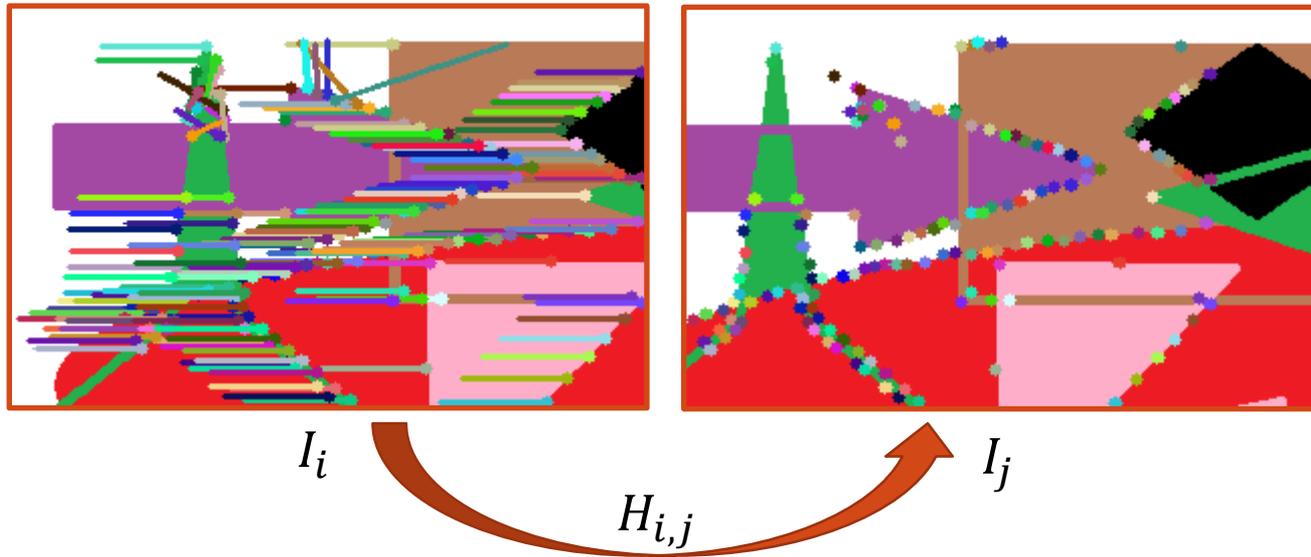
- Por DLT:

$$\begin{pmatrix} x_1^i & x_2^i & 1 & 0 & 0 & 0 & -x_1^j x_1^i & -x_1^j x_2^i & -x_1^j \\ 0 & 0 & 0 & x_1^i & x_2^i & 1 & -x_2^j x_1^i & -x_2^j x_2^i & -x_2^j \\ \vdots & & & \vdots & & & & \vdots & \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{33} \end{pmatrix} = \mathbf{0}$$

$$A_{2M \times 9} \mathbf{h}_9 = \mathbf{0}$$

- Resolver el sistema aplicando SVD
 - `>> [U, S, V] = svd(A);`
 - `>> h = V(:,ultimo); %Si los valores propios estan ordenados de mayor a menor`

CALCULAR LA HOMOGRAFÍA (C2)



Para este par de imágenes se sabe que solo hay un corrimiento de 50 píxeles hacia la izquierda en I_j respecto a I_i , entonces:

$$H_{i,j} = \begin{pmatrix} 1 & 0 & -50 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Resultado computacional:

$$H_{i,j} = \begin{pmatrix} 0.999954 & 0.000025 & -50.001379 \\ 0.000030 & 0.999853 & 0.003057 \\ -0.000000 & 0.000000 & 1.000000 \end{pmatrix}$$

ESTIMACIÓN DE LA PROBABILIDAD A POSTERIORI MÁXIMA (MAP)

- Sea \mathcal{E} un campo aleatorio de observaciones
- Sea \mathcal{V} un modelo del campo aleatorio que queremos estimar
- Sean E, v las respectivas realizaciones de \mathcal{E} y \mathcal{V}
- Para calcular v basado en E una herramienta es usar la estimación del MAP:

$$v^* = \underset{v}{\operatorname{argmax}} P(\mathcal{V} = v | E) = \underset{v}{\operatorname{argmax}} P(\mathcal{E} = E | v) P(\mathcal{V} = v)$$

Probabilidad
a posteriori

Verosimilitud

Probabilidad
a priori

Note que siguiendo Bayes, el termino $P(\mathcal{E} = E)$ no se toma en cuenta ya que no depende de v

$$P(\mathcal{V} = v | E) = \frac{P(\mathcal{E} = E | v) P(\mathcal{V} = v)}{P(\mathcal{E} = E)}$$

ESTIMACIÓN MAP PARA EL MOVIMIENTO

- Sea $P(\mathcal{J}_k = I_k | D_k = \vec{d}_k; I_{k-1})$ la relación entre I_k y I_{k-1} a través de \vec{d}_k
- Sea el error $E_k(\vec{x}) = I_k(\vec{x}) - I_{k-1}(\vec{x} - \vec{d}_k(\vec{x}))$
- Suposiciones:
 - $E_k(\vec{x})$ es una normal con media cero
 - $P(D_k = \vec{d}_k; I_{k-1})$ es una distribución Gibbs (es un MRF) definido en ciertos cliques
- Ya que tanto la verosimilitud como la probabilidad a priori pueden ser exponenciales, entonces el MAP puede ser:

$$\vec{d}_k^* = \underset{\vec{d}_k}{\operatorname{argmin}} \sum_{\vec{x}} \left\{ \left[I_k(\vec{x}) - I_{k-1}(\vec{x} - \vec{d}_k(\vec{x})) \right]^2 + \left[\sum_{\vec{y} \in N_{\vec{x}}} \left\| \vec{d}_k(\vec{x}) - \vec{d}_k(\vec{y}) \right\|_2^2 \right] \right\},$$

donde $N_{\vec{x}}$ es la vecindad de \vec{x} y $\|\cdot\|_2$ denota la distancia Euclidiana.

Podemos usar: Recocido Simulado, ICM, HCF, GraphCut, QMMF, etc.