

Flujo óptico.

Adolfo Otero

José Otero

José Antonio Corrales

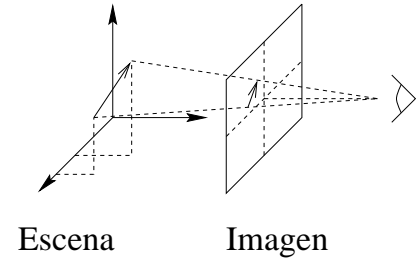
Departamento de Informática.

Area de Lenguajes y Sistemas Informáticos

Universidad de Oviedo

Definición de flujo óptico.

- Proyección en el plano de la imagen del movimiento tridimensional de los puntos de los objetos presentes en la imagen.
- Refleja los cambios en la imagen debido al movimiento de los objetos presentes en la escena.



2

Flujo óptico basado en la correlación.

Algoritmo 2D2D:

- “BMA”, block matching algorithm.
- Maximización de la correlación.

$$\phi(\delta x, \delta y, t) = I^w * I =$$

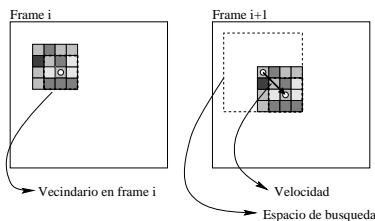
$$\int I^w(\xi, \eta, t) I(\delta x + \xi, \delta y + \eta, t + \delta t) d\xi d\eta$$

$$I^w(\xi, \eta, t) = I(\xi, \eta, t) \quad \forall (\xi, \eta) \in w$$

L_2 , es mínima cuando ϕ es máxima.

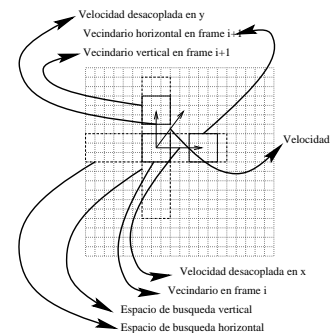
$D(\delta_x, \delta_y)$ distancia L_2 entre dos regiones de dos frames distintos centradas en (x, y) .

$S^* = (\delta_x^*, \delta_y^*)$ minimiza D $u^* = \frac{S^*}{\Delta t}$, Δt es el intervalo de adquisición de imágenes.



Algoritmo 1D2D

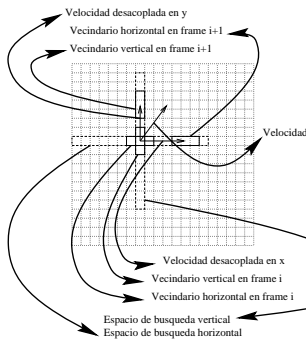
- Sustituye correlación 2D2D por una correlación 1D de regiones 2D.
- La componente de u según el eje x puede ser calculada mediante un δ_x^* que minimice $D(\delta_x, 0)$, la distancia L_2 entre las regiones en función sólo de x .
- Análogamente se podría calcular δ_y^*
- Se calculan las componentes horizontal y vertical de u por separado, reduciendo mucho el espacio de búsqueda necesario.



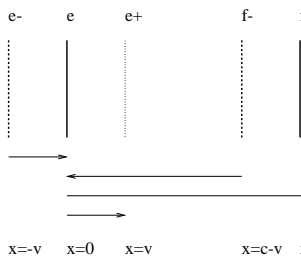
Algoritmo 1D1D

En lugar de $\phi(\delta_x, 0)$, sea una proyección sobre x de $\phi(\delta_x, \delta_y)$, obtenida mediante alguna forma de promediado en y , es decir $\phi * h_2 \Rightarrow \phi * h_2 = (I_t * h) * (I_{t+\delta t} * h)$, donde $I_t = I(x, y, t)$. Proyectar la función de correlación es equivalente a filtrar de forma apropiada las imágenes antes de efectuar la correlación.

Si se está calculando la componente vertical del flujo óptico, se realizará un filtrado horizontal de las imágenes y posteriormente una correlación de regiones 1D tomadas verticalmente de las imágenes filtradas.



Si se usan tres frames, el emparejamiento hacia atrás de e podría ser $e-$ o $f-$ con un desplazamiento de $x = v$ o $x = -(c - v)$ respectivamente. En media $x = \frac{v+v+c+v-(c-v)}{4} = v$.

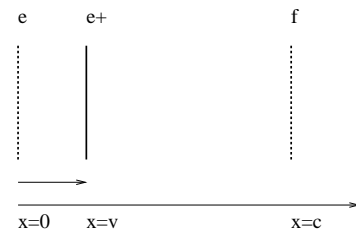


Errores en emparejamientos usando dos imágenes

Considérese un par de edgels e y f , en las posiciones $x = 0$ y $x = c$, moviéndose a lo con velocidad v en el eje x .

La posición en el siguiente frame de e será $e+$, en la que $x = v$. La posición de f será $f+$ en $x = c + v$.

Un algoritmo que use dos frames podrá emparejar e con $e+$ o con $f+$. En media $x = \frac{(v+v+c)}{2} = v + \frac{c}{2}$



Flujo óptico lineal.

El algoritmo 2D2D o BMA crece en complejidad con el cuadrado del espacio de búsqueda:

- Sea v el tamaño del vecindario, e el del espacio de búsqueda e .
- En cada dimensión (horizontal y vertical) se podrá desplazar a $e - v + 1$ posiciones posibles.
- En total se tendrán $(e - v + 1)^2$ evaluaciones de la distancia L_2 .

El tamaño del espacio de búsqueda limita la velocidad máxima medible.

$$V_{max} = \frac{c}{T}$$

Normalmente T es cte.

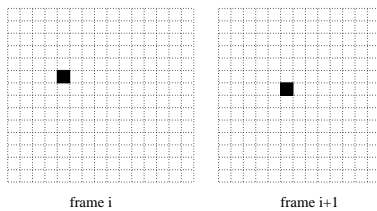
Si T es variable, se pueden tomar e y $v/e - v = 2$ y reducir el número de evaluaciones de L_2 a 9.

Siempre será posible adquirir imágenes a $T, 2T, 3T, \dots, sT$.

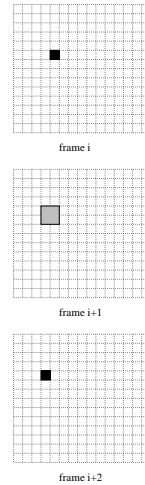
De esta forma el algoritmo 2D2D ya no crece en complejidad con el espacio de búsqueda sino que lo hace linealmente respecto al tiempo.

En la figura un pixel se desplaza en un tiempo δt una distancia $(0, -1) \Rightarrow$ velocidad = $(0, \frac{-1}{\delta t})$.

La operación se repite con distintos intervalos de tiempo y se escoge el mejor matching como desplazamiento verdadero.



Al tomar el mejor matching en el tiempo, se obvia el problema del aliasing espacial introducido por la naturaleza discreta de los sensores CCD.



Aunque los desplazamientos sólo pueden ser $\{-1, 0, 1, \}$ como las velocidades se obtienen dividiendo esos desplazamientos entre el tiempo de exploración, se pueden obtener velocidades menores de 1 pixel/frame.

El problema de la apertura en el dominio del tiempo

En el dominio temporal ocurre cuando se toma una ventana temporal demasiado pequeña.

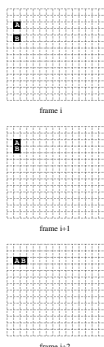
Velocidad de A $(\frac{1}{2T}, \frac{-1}{2T})$.

Velocidad de B $(1, \frac{1}{T})$.

Matching equivocado entre el segundo y tercer frames:

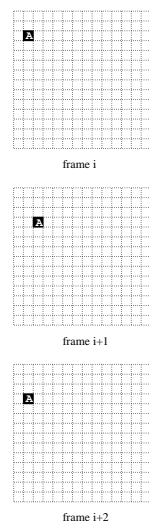
Velocidad de B $(\frac{1}{2T}, \frac{1}{2T})$.

Velocidad de A $(0, \frac{-1}{T})$.



Aliasing temporal. Se da cuando la ventana temporal es demasiado grande y los pixels retroceden.

En este caso el algoritmo devuelve una velocidad menor que la correcta.



Flujo óptico mediante LUT.

- Cálculo off-line del FO correspondiente a parejas de vecindario y espacio de búsqueda.
- Los resultados se guardan en una LUT indexada por las intensidades de ambos.
 - Se concatenan los valores de intensidad para formar números en base n.
- Para que la conversión de valores sea más rápida los números se convierten a decimal mediante un árbol.
- Hace innecesarios los cálculos en coma flotante.
- Es necesario reducir la profundidad de la intensidad.
- Es necesario reducir la longitud de los números \Rightarrow algoritmo 1D1D.

Identificación de vecindarios como números en base n.

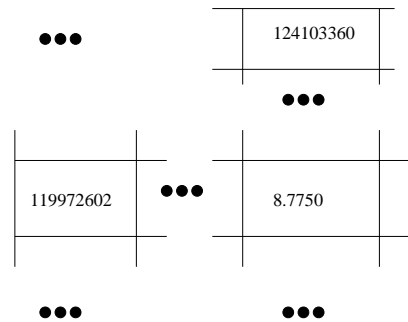
7	1	1
5	2	1
3	7	2

Octal: 711521372
Decimal: 119972602

7	3	1
3	2	5
3	0	0

Octal: 731325300
Decimal: 124103360

Cálculo de la distancia L_2 mediante LUT.



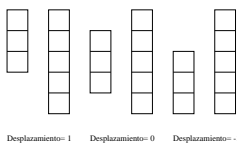
Obtención del promediado horizontal a partir de una L.U.T.

	A	B	1	

Hexadecimal: AB1
Decimal: 2737

2737	7.33
------	------

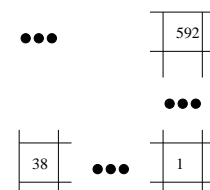
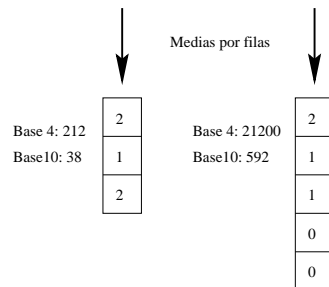
Convenio de signos



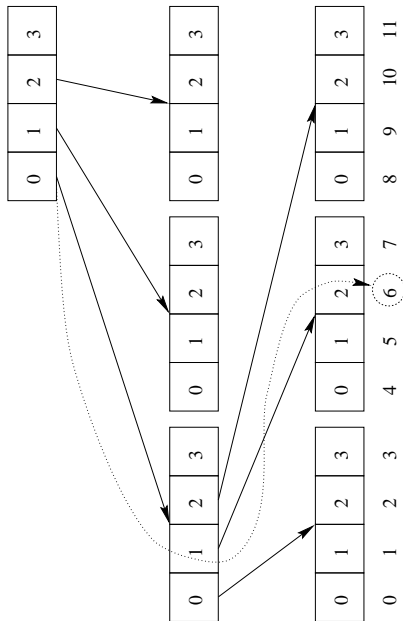
Cálculo de una componente del flujo óptico mediante tablas L.U.T.

0	2	3
1	0	3
2	1	2

0	2	3	0	0
1	0	3	0	0
2	1	2	0	0
0	0	0	0	0
0	0	0	0	0



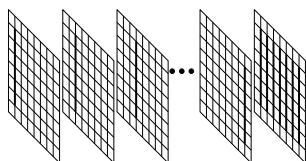
Conversión a base 10.



Flujo óptico basado en la transformada de Fourier.

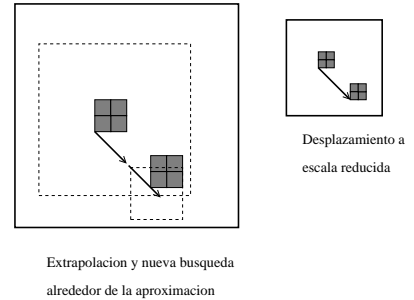
La secuencia se entiende como una señal variable con el tiempo.

- Análisis de la respuesta de distintos filtros.
 - Sintonizados con distintas velocidades.
 - Con distinta orientación.
- El filtro con máxima respuesta identifica el módulo y la dirección de la velocidad.



Mejoras a los algoritmos anteriores

- En cuanto a precisión: interpolación de la imagen para obtener desplazamientos < 1 pixel, combinación de varios métodos.
- En cuanto a eficiencia: implementación jerárquica.



Flujo óptico basado en derivadas espaciotemporales.

- Parte de suponer que la intensidad de los puntos de los objetos de la escena no varían con el tiempo.
- Por lo tanto los cambios en la intensidad los pixels de la imagen se deben exclusivamente al movimiento.
- O.F.C. (Optical Flow Constraint):

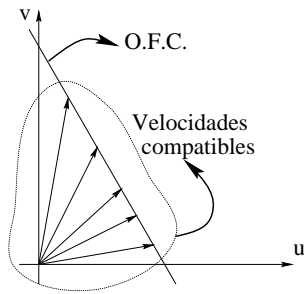
$$\frac{df(x, y, t)}{dt} = 0$$

La O.F.C. desarrollada conduce a:

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} = \frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v = \nabla(f) \cdot \vec{c}$$

La ecuación anterior relaciona la velocidad de un pixel con las derivadas parciales de la intensidad del mismo pixel respecto a x, y, t .

- En dicha ecuación aparecen dos incógnitas (u, v), no permite calcular de forma total la velocidad.
 - “Problema de la apertura”: imposibilidad de calcular el flujo óptico sólo con información local.
 - Múltiples velocidades satisfacen la O.F.C.



21

Velocidad normal.

A partir de una sola OFC sólo se puede determinar la velocidad en la dirección del gradiente:

Velocidad normal.

$$u_n = \frac{\frac{\partial f}{\partial t}}{\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}} \quad (1)$$

$$\theta = \tan^{-1} \left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \quad (2)$$

u_n es el módulo de la velocidad y θ el ángulo de la dirección del gradiente con el eje de abscisas positivo.

Regularización. Algoritmo de Horn y Schunck.

- Impone una condición adicional al proceso de cálculo.
- Aporta información global.
- Maximiza la suavidad del flujo óptico minimizando:

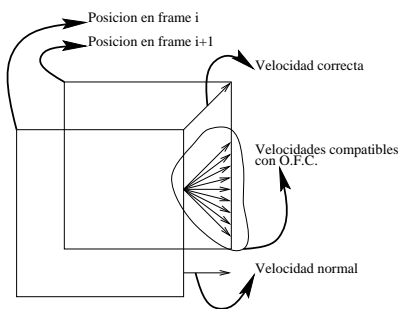
$$\varepsilon_c = \sqrt{(\bar{u} - u)^2 + (\bar{v} - v)^2}$$

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = k(\bar{u} - u)$$

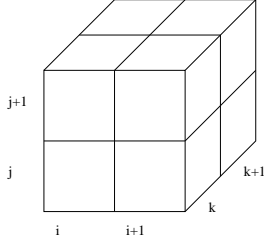
$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = k(\bar{v} - v)$$

- Minimiza el error respecto de la O.F.C.

$$\varepsilon_b = \frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v + \frac{\partial f}{\partial t}$$



Cálculo de las derivadas.



$$\frac{\partial f}{\partial x}(i, j, k) = \frac{1}{\delta x}(f(i+1, j, k) + f(i+1, j, k+1) + f(i+1, j+1, k) + f(i+1, j+1, k+1)) - \frac{1}{\delta x}(f(i, j, k) + f(i, j, k+1) + f(i, j+1, k) + f(i, j+1, k+1))$$

$$\frac{\partial f}{\partial y}(i, j, k) = \frac{1}{\delta y}(f(i, j+1, k) + f(i, j+1, k+1) + f(i+1, j+1, k) + f(i+1, j+1, k+1)) - \frac{1}{\delta y}(f(i, j, k) + f(i, j, k+1) + f(i+1, j, k) + f(i+1, j, k+1))$$

$$\frac{\partial f}{\partial t}(i, j, k) = \frac{1}{\delta t}(f(i, j, k+1) + f(i, j+1, k+1) + f(i+1, j, k+1) + f(i+1, j+1, k+1)) - \frac{1}{\delta t}(f(i, j, k) + f(i, j+1, k) + f(i+1, j, k) + f(i+1, j+1, k))$$

$$\bar{u}_{i,j,k} = \frac{1}{6}(u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}) + \frac{1}{12}(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k})$$

$$\bar{v}_{i,j,k} = \frac{1}{6}(v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}) + \frac{1}{12}(v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k})$$

El determinante de la matriz de coeficientes del sistema es:

$$\alpha^2 \left(\alpha^2 + \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right)$$

La solución del sistema es:

$$\left(\alpha^2 + \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) u =$$

$$\left(\alpha^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) \bar{u} - \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \bar{v} - \frac{\partial f}{\partial x} \frac{\partial f}{\partial t}$$

$$\left(\alpha^2 + \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) v =$$

$$-\frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \bar{u} + \left(\alpha^2 + \left(\frac{\partial f}{\partial x} \right)^2 \right) \bar{v} - \frac{\partial f}{\partial y} \frac{\partial f}{\partial t}$$

La expresión a minimizar donde se pondera el efecto de la suavidad del flujo óptico y el ajuste a la OFC es:

$$\varepsilon^2 = \alpha^2 \varepsilon_c^2 + \varepsilon_b^2$$

Derivando respecto a u y v :

$$\frac{\partial \varepsilon^2}{\partial u} = -2\alpha^2(\bar{u} - u) + 2 \left(\frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v + \frac{\partial f}{\partial t} \right) \frac{\partial f}{\partial x}$$

$$\frac{\partial \varepsilon^2}{\partial v} = -2\alpha^2(\bar{v} - v) + 2 \left(\frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v + \frac{\partial f}{\partial t} \right) \frac{\partial f}{\partial y}$$

Igualando las anteriores ecuaciones a cero se llega al sistema de dos ecuaciones y dos incógnitas:

$$\left(\alpha^2 + \left(\frac{\partial f}{\partial x} \right)^2 \right) u + \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} v = \alpha^2 \bar{u} - \frac{\partial f}{\partial x} \frac{\partial f}{\partial t}$$

$$\frac{\partial f}{\partial x} \frac{\partial f}{\partial y} u + \left(\alpha^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) v = \alpha^2 \bar{v} - \frac{\partial f}{\partial y} \frac{\partial f}{\partial t}$$

\bar{u} y \bar{v} son las medias de las componentes horizontal y vertical de la velocidad en un entorno 3x3 del píxel a analizar, ponderada según la distancia al píxel central.

1/12	1/6	1/12
1/6		1/6
1/12	1/6	1/12

Algoritmo iterativo de Horn y Schunck para el cálculo de flujo óptico usando un par de imágenes:

1. Inicializar $c(i, j) = (0, 0)$ para todo (i, j) .
2. Sea k la iteración actual, calcular u^k y v^k para todos los píxeles (i, j) evaluar:

$$u^k(i, j) = \bar{u}^{k-1}(i, j) - f_x(i, j) \frac{P(i, j)}{D(i, j)}$$

$$v^k(i, j) = \bar{v}^{k-1}(i, j) - f_y(i, j) \frac{P(i, j)}{D(i, j)}$$

donde $P = f_x \bar{u} + f_y \bar{v}$, $D = \alpha^2 + f_x^2 + f_y^2$.

3. Parar si $\sum_i \sum_j E^2(i, j) < \varepsilon$, en caso contrario volver a 2.

Inconvenientes del algoritmo de Horn y Schunck:

- Uso de derivadas \Rightarrow es necesario filtrar imágenes para paliar efecto del ruido.
- Ponderación suavidad/OFC?
- Condición de regularización: falla en contornos, algoritmo no converge.

Mejoras:

- Procesamiento de varias imágenes: utilizar velocidades entre el par anterior como inicio del cálculo del siguiente par.
- Similar al anterior: implementación jerárquica.

Derivadas de segundo orden.

- Suposición adicional: $\frac{d\nabla f(x,y,t)}{dt} = 0$.
- Sistema de ecuaciones lineales sobredeterminado:

$$\begin{aligned} \frac{\partial f(x,y,t)}{\partial x}u + \frac{\partial f(x,y,t)}{\partial y}v &= -\frac{\partial f(x,y,t)}{\partial t} \\ \frac{\partial^2 f(x,y,t)}{\partial x^2}u + \frac{\partial^2 f(x,y,t)}{\partial x\partial y}v &= -\frac{\partial^2 f(x,y,t)}{\partial x\partial t} \\ \frac{\partial^2 f(x,y,t)}{\partial x\partial y}u + \frac{\partial^2 f(x,y,t)}{\partial y^2}v &= -\frac{\partial^2 f(x,y,t)}{\partial y\partial t} \end{aligned}$$

Estimación sobre el vecindario.

- Resuelve el sistema formado por O.F.C. de pixels vecinos.
- Supone que pixels próximos se mueven con velocidades similares.
- Realiza una estimación sobre el total de los datos.
- Los datos forman un sistema sobredeterminado.

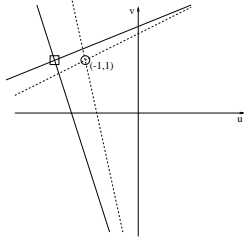
$$\begin{aligned} \frac{\partial f(x_1,y_1,t)}{\partial x}u + \frac{\partial f(x_1,y_1,t)}{\partial y}v &= -\frac{\partial f(x_1,y_1,t)}{\partial t} \\ &\dots \\ \frac{\partial f(x_n,y_n,t)}{\partial x}u + \frac{\partial f(x_n,y_n,t)}{\partial y}v &= -\frac{\partial f(x_n,y_n,t)}{\partial t} \end{aligned}$$

Inconvenientes de las alternativas anteriores.

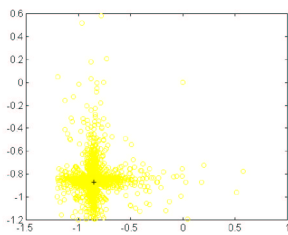
- Regularización: F.O. no necesariamente "verdadero", se imponen condiciones "a priori".
- Derivadas de segundo orden: imprecisión y condiciones adicionales.
- Estimación sobre un conjunto de datos crudo: puede conducir a una estimación que puede estar desviada de la correcta.
- Correlación y transformada de Fourier: costosas computacionalmente.

Efecto del ruido en la estimación.

- El corte de O.F.C afectadas por el ruido devuelve una estimación desviada.

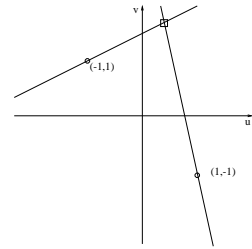


- Una estimación por mínimos cuadrados sólo sería fructífera si el ruido fuese isotrópico.

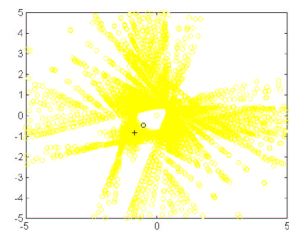


Efecto de la oclusión.

- La intersección de O.F.C. correspondientes a objetos distintos produce una velocidad inexistente.



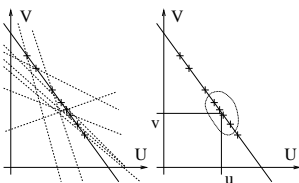
- El efecto agregado puede producir una desviación importante en la estimación.



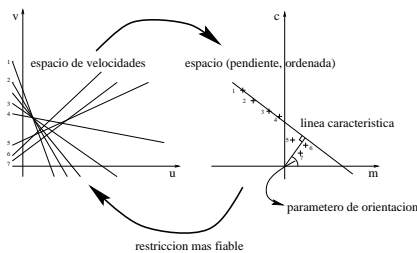
Estimación robusta del flujo óptico.

Existen trabajos en los que se propone el análisis de la O.F.C correspondiente al pixel central relacionado las O.F.C de cada vecino.

- Schunck propone un clustering unidimensional de los cortes de las O.F.C.



- Nesi y otros proponen un clustering de O.F.C alrededor de la O.F.C. central con el fin de determinar la O.F.C más fiable.



Problemas de estas alternativas.

- Algoritmo de Schunck:
 - Sensible al error en la O.F.C. central.
 - Clustering eficiente computacionalmente a costa de reducir su robustez.
- Nesi et al:
 - Clustering de O.F.C conduciendo al cálculo correspondiente a la velocidad normal, no al flujo óptico completo.
 - Proceso sensible a errores en la O.F.C. correspondiente al pixel central.

Medidas para evitar el error procedente de los outliers.

- Estimación sobre el mayor número de datos posible: $n^2 - 1$ frente a $n^2 * (n^2 - 1) / 2$ utilizando el mismo vecindario.
- No depender totalmente de la información que se desprende del pixel central.
- Análisis de la distribución bidimensional de los cortes de O.F.C., con el fin de identificar el cluster correspondiente al movimiento dominante en el vecindario analizado.
- Utilización de estimadores robustos de tendencia central.

Existen otros estimadores de la función de densidad más adecuados.

- Estimación Naif: asigna a cada observación el número de observaciones que existen en un entorno.

$$\hat{f}(x) = \frac{1}{2hn} [\text{número de } X_i \in (x - h, x + h)]$$

- Estimación Kernel: similar a la anterior pero otorgando peso decreciente a las observaciones al alejarse de la observación central.

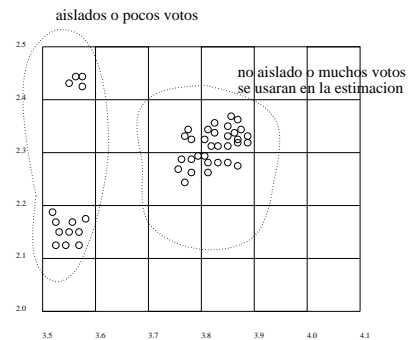
$$\int_{-\infty}^{\infty} K(x) dx = 1$$

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - X_i}{h}\right)$$

- K-vecinos.

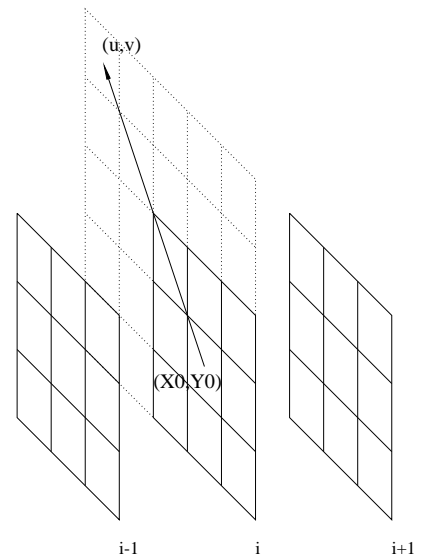
Histograma como estimación de la función de densidad.

- Si se calcula el histograma de la distribución bidimensional de las velocidades, la moda se puede hallar buscando el máximo de la distribución.
- Debido a la naturaleza del histograma, es necesario realizar operaciones que eliminen picos aislados o con pocos votos.
- El resultado es un algoritmo satisfactorio pero ineficiente.

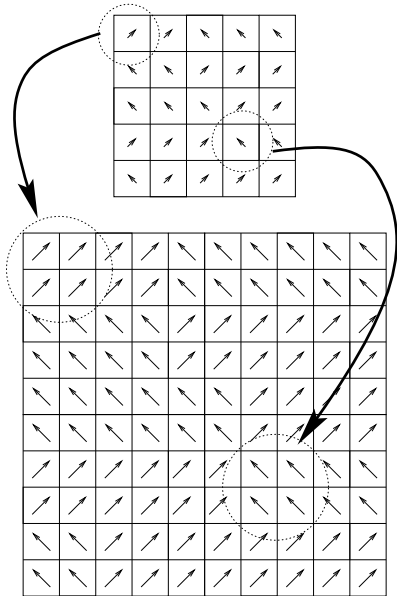


Inconsistencia en la estimación de las derivadas temporales.

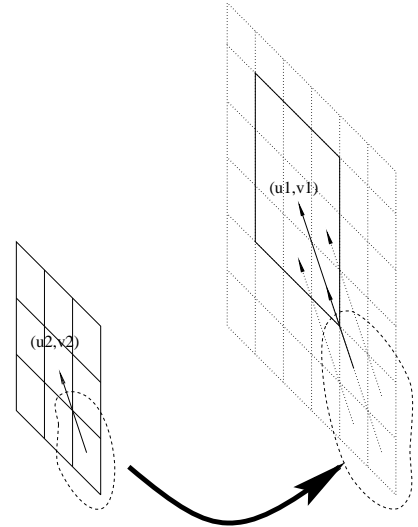
Ocurre cuando los desplazamientos de los pixels son tales que sobrepasan los límites del vecindario utilizado en la estimación.



Una estimación a menor escala se puede extender a una escala superior.



La estimación grosera puede refinarse en una escala superior.

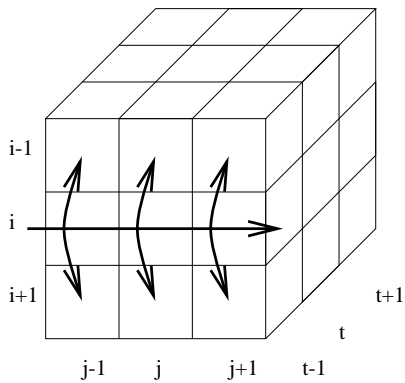


Cálculo de derivadas.

- Diferencias centrales.

$$f'(x) \simeq \frac{8(f(x + \delta) - f(x - \delta)) - (f(x + 2\delta) - f(x - 2\delta))}{12\delta x}$$

- Promediado en pixels adyacentes en dirección normal a la de la derivada.



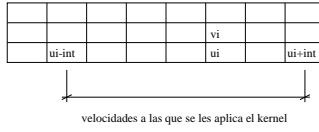
Sistemas de OFCs descartados a priori.

- Sistemas cuyo discriminante sea menor que un umbral. Por ser más sensibles al ruido.
- Sistemas cuya solución exceda un umbral. Debido a un conocimiento a priori de la escena.

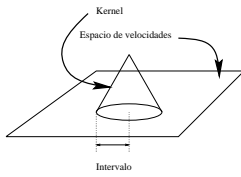
$$x = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}}$$

Estimación de función de densidad y moda.

- Velocidades ordenadas en un array según la componente u y v .



- El kernel no es aplicado a las componentes que caen fuera del intervalo en el que este es mayor que cero a priori.

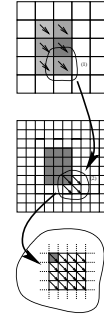


$$d = \sqrt{(u - u_i)^2 + (v - v_i)^2}$$

$$k(u, v) = \begin{cases} 1 - \frac{d}{Intervalo} & \text{si } d < Intervalo \\ 0 & \text{en caso contrario} \end{cases}$$

Efecto de bloque.

- Se produce al propagarse la estimación grosera a mayor resolución.



- Es necesario una medida de la bondad que permita descartar la estimación grosera cuando no es adecuada.
- El número de puntos que han colaborado en la estimación es una medida de la bondad.