



Ciencia y Tecnología

Secretaría de Ciencia, Humanidades, Tecnología e Innovación



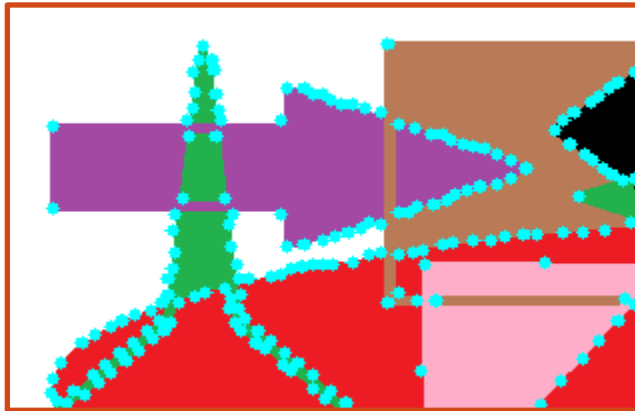
CIMAT
UNIDAD MÉRIDA

PUNTOS DE INTERÉS Y DESCRIPTORES

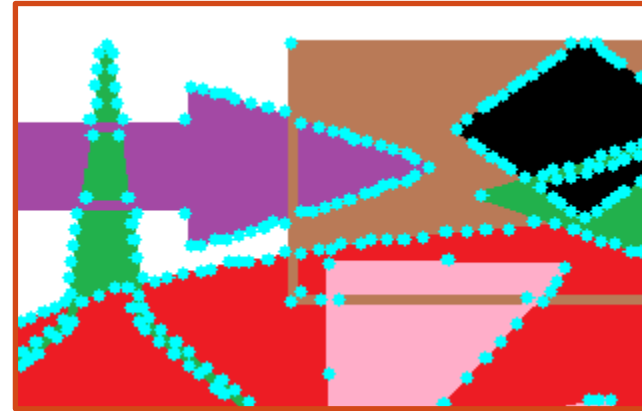
Dr. Francisco J. Hernández López
SECIHTI – CIMAT-Mérida
fcoj23@cimat.mx, www.cimat.mx/~fcoj23



DETECCIÓN DE CARACTERÍSTICAS



I_i



I_j

Detectores
Harris, Shi-Tomasi, etc.

$$\begin{aligned}\vec{c}_1 &= (x_1, y_1) \\ \vec{c}_2 &= (x_2, y_2) \\ &\vdots \\ \vec{c}_m &= (x_m, y_m)\end{aligned}$$

Descriptores
SIFT, SURF, ORB, etc.

$$\begin{aligned}\vec{c}_1 &= (d_1^1, d_1^2, \dots, d_1^n) \\ \vec{c}_2 &= (d_2^1, d_2^2, \dots, d_2^n) \\ &\vdots \\ \vec{c}_m &= (d_m^1, d_m^2, \dots, d_m^n)\end{aligned}$$

DETECTOR DE ESQUINAS DE HARRIS

- Para cada pixel $\vec{x} = (x, y)$

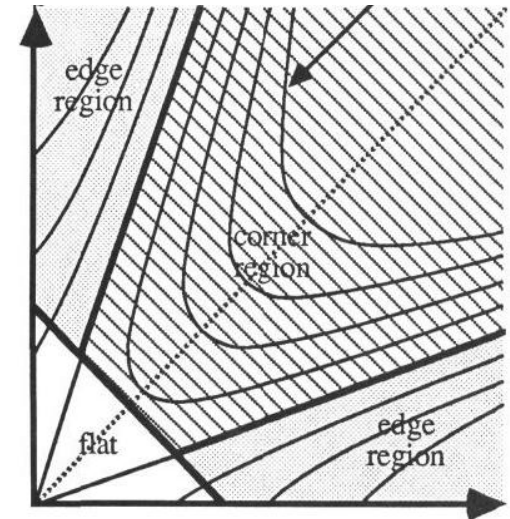
1. Calcular la matriz

$$M(\vec{x}) = \begin{bmatrix} \sum_{\vec{y} \in W_{\vec{x}}} \left(\frac{\partial}{\partial x} I(\vec{y}) \right)^2 & \sum_{\vec{y} \in W_{\vec{x}}} \left(\frac{\partial}{\partial x} I(\vec{y}) \right) \left(\frac{\partial}{\partial y} I(\vec{y}) \right) \\ \sum_{\vec{y} \in W_{\vec{x}}} \left(\frac{\partial}{\partial x} I(\vec{y}) \right) \left(\frac{\partial}{\partial y} I(\vec{y}) \right) & \sum_{\vec{y} \in W_{\vec{x}}} \left(\frac{\partial}{\partial y} I(\vec{y}) \right)^2 \end{bmatrix}$$

Con $W_{\vec{x}}$ una ventana centrada en $\vec{x}$. Harris y Stephens proponen que $W_{\vec{x}}$ sea una ventana Gaussiana.

2. Los eigenvalores λ_1 y λ_2 de M son proporcionales a las curvaturas de I :

- λ_1 y λ_2 pequeños $\rightarrow$ Región uniforme
- $\lambda_1 \gg \lambda_2$ o $\lambda_1 \ll \lambda_2$ $\rightarrow$ Hay un borde
- λ_1 y λ_2 grandes $\rightarrow$ Hay una esquina



Harris, Christopher G., and Mike Stephens. "A combined corner and edge detector." *Alvey vision conference*. Vol. 15. No. 50. 1988.

Puntos de Interés y Descriptores. Francisco J. Hernández-López

Ene-Jun 2025

DETECTOR DE ESQUINAS DE HARRIS (C1)

$$c(\vec{x}) = \lambda_1 \lambda_2 - k (\lambda_1 + \lambda_2)^2$$

$$= \det(M(\vec{x})) - k \left(\text{traza}(M(\vec{x})) \right)^2$$

Puede ser expresado sin necesidad de calcular los eigenvalores

k es un parámetro constante que se determina de forma empírica, valores comúnmente usados están en el rango $[0.04, 0.15]$.

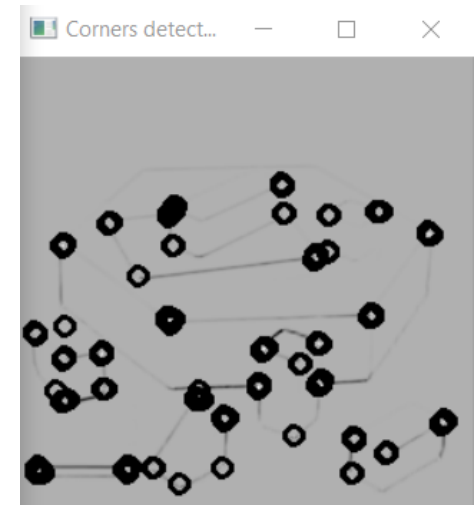
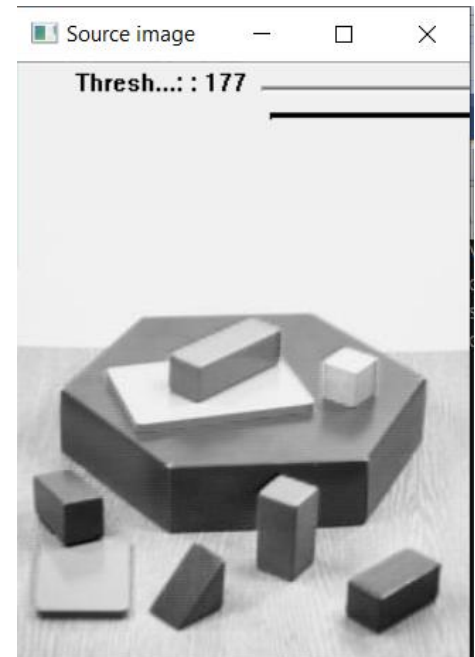
3. Filtrar los valores de $c(\vec{x})$ no máximos en una vecindad $\mathcal{N}_{\vec{x}}$ (por ej, de 8 vecinos alrededor de $\vec{x}$)

Si $c(\vec{x}) < \theta \max_{\vec{y} \in \mathcal{N}_{\vec{x}}} \{c(\vec{y})\}$ entonces eliminamos el punto

θ es un porcentaje del máximo valor encontrado.

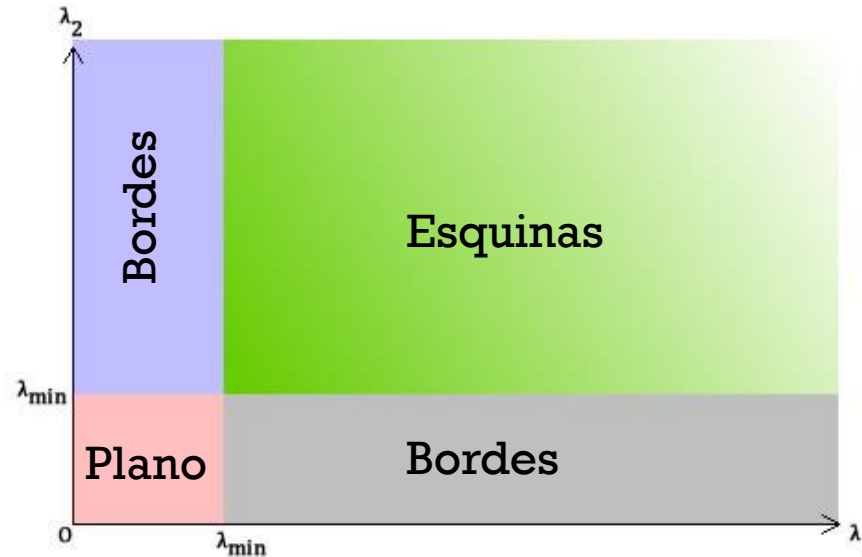
EJEMPLO USANDO OPENCV

```
57 void cornerHarris_demo( int, void* )
58 {
59     /// Detector parameters
60     int blockSize = 2;
61     int apertureSize = 3;
62     double k = 0.04;
63
64     /// Detecting corners
65     Mat dst = Mat::zeros( src.size(), CV_32FC1 );
66     cornerHarris( src_gray, dst, blockSize, apertureSize, k );
67
68     /// Normalizing
69     Mat dst_norm, dst_norm_scaled;
70     normalize( dst, dst_norm, 0, 255, NORM_MINMAX, CV_32FC1, Mat() );
71     convertScaleAbs( dst_norm, dst_norm_scaled );
72
73     /// Drawing a circle around corners
74     for( int i = 0; i < dst_norm.rows ; i++ )
75     {
76         for( int j = 0; j < dst_norm.cols; j++ )
77         {
78             if( (int) dst_norm.at<float>(i,j) > thresh )
79             {
80                 circle( dst_norm_scaled, Point(j,i), 5, Scalar(0), 2, 8, 0 );
81             }
82         }
83     }
84
85     /// Showing the result
86     namedWindow( corners_window );
87     imshow( corners_window, dst_norm_scaled );
88 }
```



SHI-TOMASI “GOOD FEATURES TO TRACK”

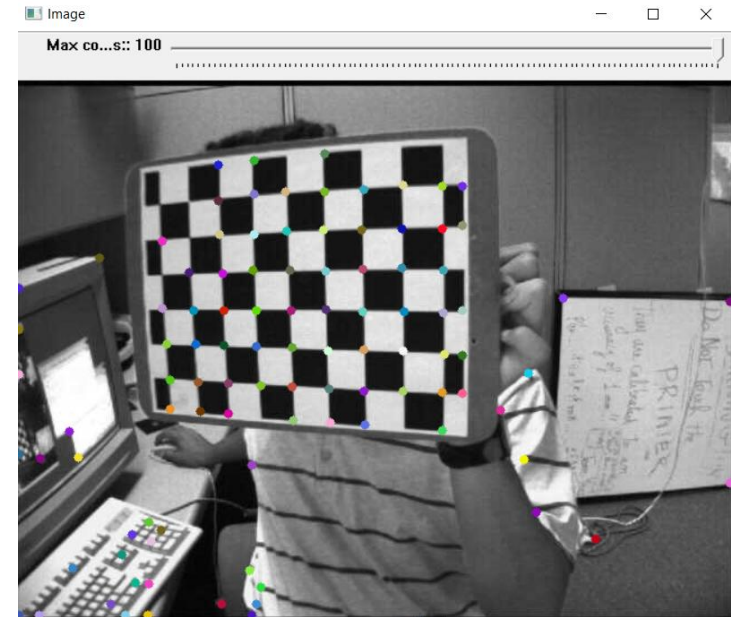
- Calcular los eigenvalores λ_1 y λ_2 de la matriz M (como en Harris)
- Un pixel $\vec{x}$ es aceptado como punto de interés si cumple:
$$c(\vec{x}) = \min(\lambda_1, \lambda_2) > \lambda$$



<http://aishack.in/tutorials/shitomasi-corner-detector/>

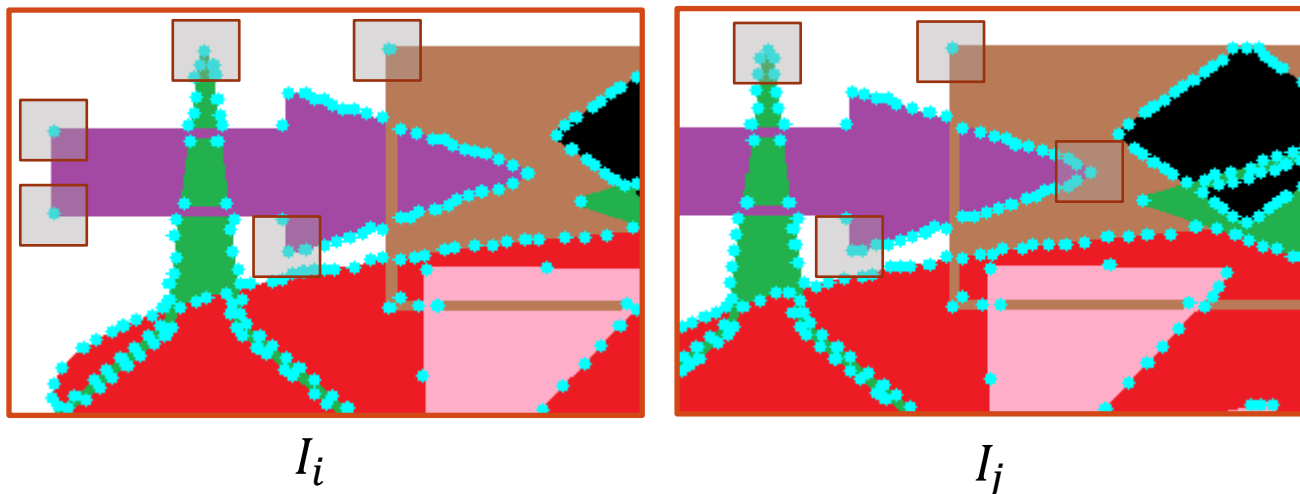
EJEMPLO USANDO OPENCV

```
61 void goodFeaturesToTrack_Demo( int, void* )
62 {
63     /// Parameters for Shi-Tomasi algorithm
64     maxCorners = MAX(maxCorners, 1);
65     vector<Point2f> corners;
66     double qualityLevel = 0.01;
67     double minDistance = 10;
68     int blockSize = 3, gradientSize = 3;
69     bool useHarrisDetector = false;
70     double k = 0.04;
71
72     /// Copy the source image
73     Mat copy = src.clone();
74
75     /// Apply corner detection
76     goodFeaturesToTrack( src_gray, corners, maxCorners,
77                         qualityLevel, minDistance, Mat(), blockSize,
78                         gradientSize, useHarrisDetector, k );
79
80     /// Draw corners detected
81     cout << "*** Number of corners detected: " << corners.size() << endl;
82     int radius = 4;
83     for( size_t i = 0; i < corners.size(); i++ )
84     {
85         circle( copy, corners[i], radius, Scalar(rng.uniform(0,255),
86             rng.uniform(0, 256), rng.uniform(0, 256)), FILLED );
87     }
88
89     /// Show what you got
90     namedWindow( source_window );
91     imshow( source_window, copy );
92 }
93 }
```



DESCRIPTORES DE CARACTERÍSTICAS

- Usados para identificar las características (puntos de interés o esquinas) en una imagen
- Con esto se podrían comparar los descriptores en diferentes imágenes y encontrar las correspondencias.
- Un descriptor podría ser una región de la imagen centrada en la característica.



SCALE INVARIANT FEATURE TRANSFORM (SIFT)

- Detecta posiciones usando extremos (máximos y mínimos) en espacio-escala en diferentes funciones Gaussianas convolucionadas con la imagen.
- Se construye un histograma de gradientes (**HOG**) a partir de magnitudes del gradiente local en las posiciones seleccionadas.
- La salida del SIFT es un vector de **128 valores** (arreglos de histogramas de 4x4 con 8 orientaciones) para cada punto detectado.

Lowe, David G. "Object recognition from local scale-invariant features." *iccv*. Ieee, 1999.

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

SIFT (C1)

1. Calcular las diferencias de Gaussianas (**DoG**)

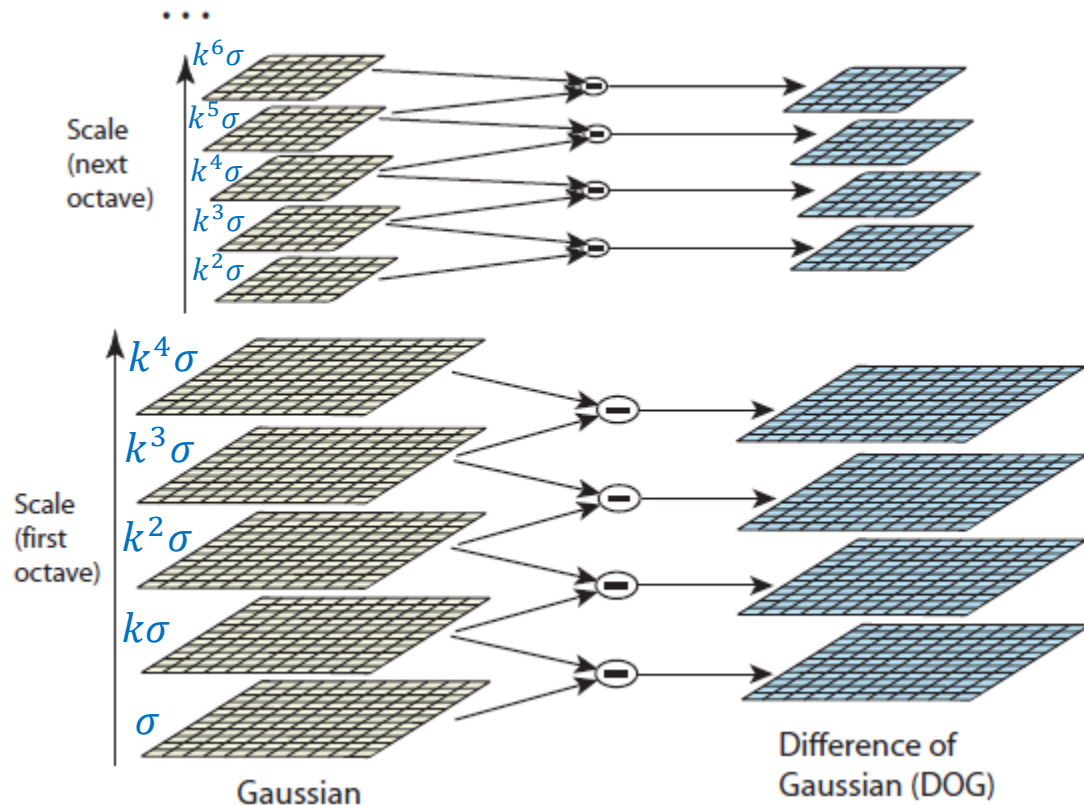
$$L(\vec{x}, \sigma) = G(\vec{x}, \sigma) \otimes I(\vec{x}), \text{ con } G(\vec{x}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(\|\vec{x}\|^2)/2\sigma^2}$$

[Lowe, 99] propone calcular extremos en diferencias de las funciones $L(\vec{x}, \sigma)$ a diferentes escalas:

$$\begin{aligned} D(\vec{x}, \sigma) &= (G(\vec{x}, k\sigma) - G(\vec{x}, \sigma)) \otimes I(\vec{x}) \\ &= L(\vec{x}, k\sigma) - L(\vec{x}, \sigma) \end{aligned}$$

Cada octava se divide en un número $s + 3$ de imágenes y $k = 2^{1/s}$

Proponen usar 4 octavas y 5 niveles de suavidad, $s = 2$ y $k = \sqrt{2}$



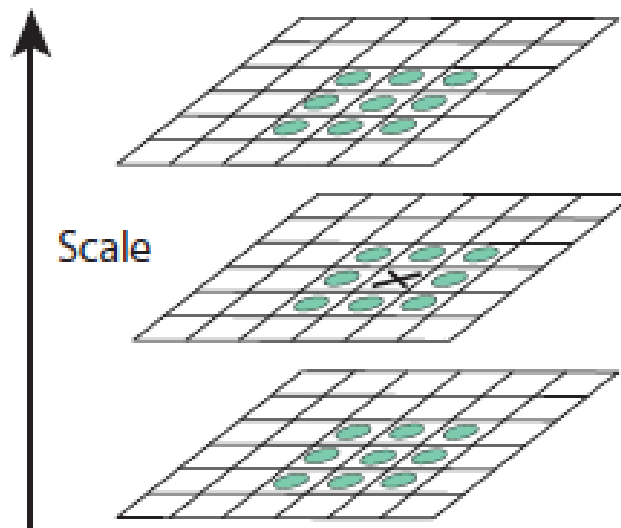
[Lowe, 2004]

SIFT (C2)

2. Detección de extremos (máximos y mínimos)

Cada punto en las imágenes $D(\vec{x}, \sigma)$, es comparado con sus 8 vecinos en la imagen visitada y con los 9 vecinos en las imágenes de escalas adyacentes.

Se seleccionan los puntos en donde la D sea más grande o más pequeña que todos sus vecinos (se requieren 26 comparaciones).



[Lowe, 2004]

SIFT (C3)

3. Localizar el punto de interés (Eliminando los de bajo contraste)

Interpolar los máximos o mínimos usando la siguiente función cuadrática (a partir de la serie de Taylor de segundo orden):

$$D(\vec{z} + \vec{\delta z}) \approx D(\vec{z}) + \left(\frac{\partial D}{\partial \vec{\delta z}} \bigg|_{\vec{z}} \right)^T \vec{\delta z} + \frac{1}{2} \vec{\delta z}^T \left(\frac{\partial^2 D}{\partial \vec{\delta z}^2} \bigg|_{\vec{z}} \right) \vec{\delta z} \quad (1)$$

con $\vec{z} = (x, y, \sigma)^T$ y $\vec{\delta z} = (\delta x, \delta y, \delta \sigma)^T$

Derivando (1) c.r.a. $\vec{\delta z}$ e igualando a cero obtenemos:

$$\vec{\delta z}^* = - \left(\frac{\partial^2 D}{\partial \vec{\delta z}^2} \bigg|_{\vec{z}} \right)^{-1} \left(\frac{\partial D}{\partial \vec{\delta z}} \bigg|_{\vec{z}} \right) \quad (2)$$

Sustituyendo (2) en (1):

$$D(\vec{z} + \vec{\delta z}^*) = D(\vec{z}) + \frac{1}{2} \frac{\partial D}{\partial \vec{\delta z}} \vec{\delta z}^* \quad (3)$$

El valor de $D(\vec{z} + \vec{\delta z}^*)$ se utiliza para rechazar extremos de bajo contraste:
Si $|D(\vec{z} + \vec{\delta z}^*)| < 0.03$ entonces se rechaza el punto.

SIFT (C4)

3. Localizar el punto de interés (Eliminando los que estén en los bordes)

La función D tendrá respuestas fuertes en los bordes de la imagen

A partir de la matriz Hessiana:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Las derivadas son estimadas tomando las diferencias entre los vecinos de los puntos candidatos.

Sean α y β los eigenvalores propios más grande y más pequeño de H , entonces:

$$\text{Traza}(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

Sea r el radio entre los eigenvalores, tal que $\alpha = r\beta$, entonces:

$$\frac{(\text{Traza}(H))^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r}$$

Por Harris sabemos
que si $\alpha \gg \beta$ o $\alpha \ll \beta$
→ Hay un borde

Si se cumple
$$\frac{(\text{Traza}(H))^2}{\text{Det}(H)} > \frac{(r + 1)^2}{r}$$

entonces se rechaza el
punto, con $r = 10$

SIFT (C5)

4. Asignar orientaciones → Para que el algoritmo sea invariante a rotaciones

Se calculan las derivadas, la magnitud del gradiente y la orientación de $L(\vec{x}, \sigma)$ en la escala σ del punto de interés $\vec{x} = (x, y)$, de la siguiente forma:

$$L_x(x, y) = L(x + 1, y) - L(x - 1, y), \quad L_y(x, y) = L(x, y + 1) - L(x, y - 1)$$

$$m(x, y) = \sqrt{(L_x(x, y))^2 + (L_y(x, y))^2}$$

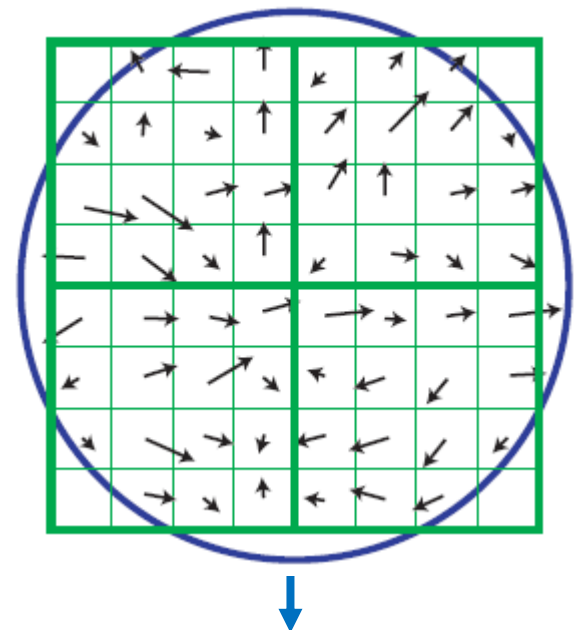
$$\theta(x, y) = \text{tg}^{-1} \left(\frac{L_y(x, y)}{L_x(x, y)} \right)$$

Crear un histograma de orientaciones pesadas en una vecindad del punto de interés

Cada muestra que se agrega al histograma, se pesa por su respectiva magnitud del gradiente y una ventana circular Gaussiana con $\sigma = 1.5\hat{\sigma}$

$\hat{\sigma} \rightarrow$ escala encontrada en el punto de interés

Vecindad de 8×8



Ventana circular Gaussiana

[Lowe, 2004]

Ene-Jun 2025

SIFT (C6)

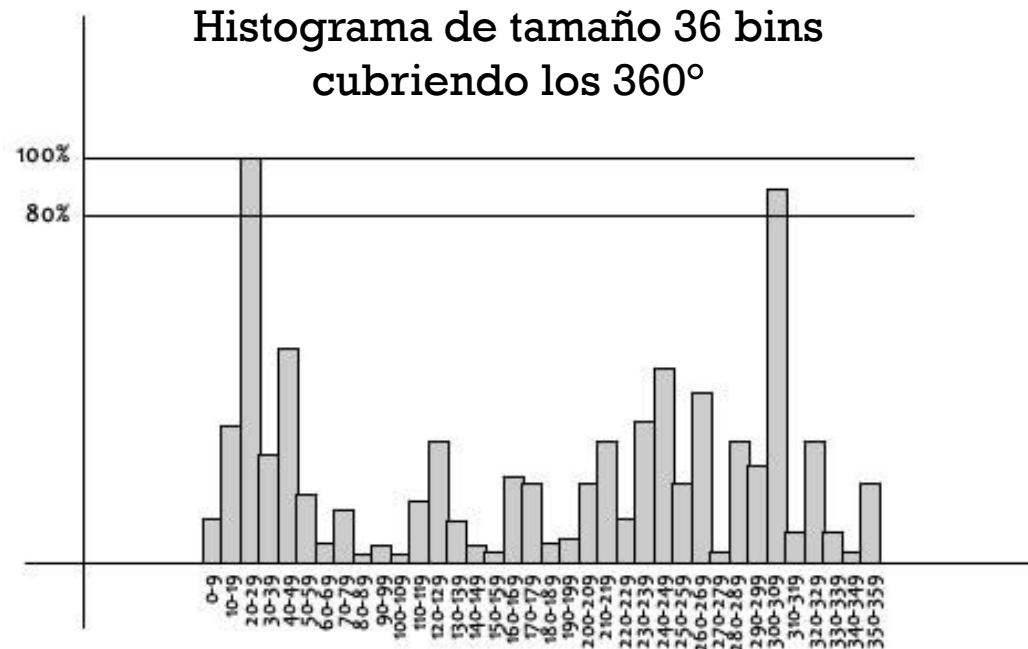
4. Asignar orientaciones

Los picos en el hist. corresp. a orientaciones dominantes de los gradientes locales

Se detecta el pico más alto y entonces se crea un punto de interés con esa orientación

Si hay más picos dentro del 80% del pico más alto, estos también se eligen para crear nuevos puntos de interés

Luego se ajusta una parábola a los 3 valores del histograma más cercanos a cada pico para interpolar la posición del pico para una mejor exactitud

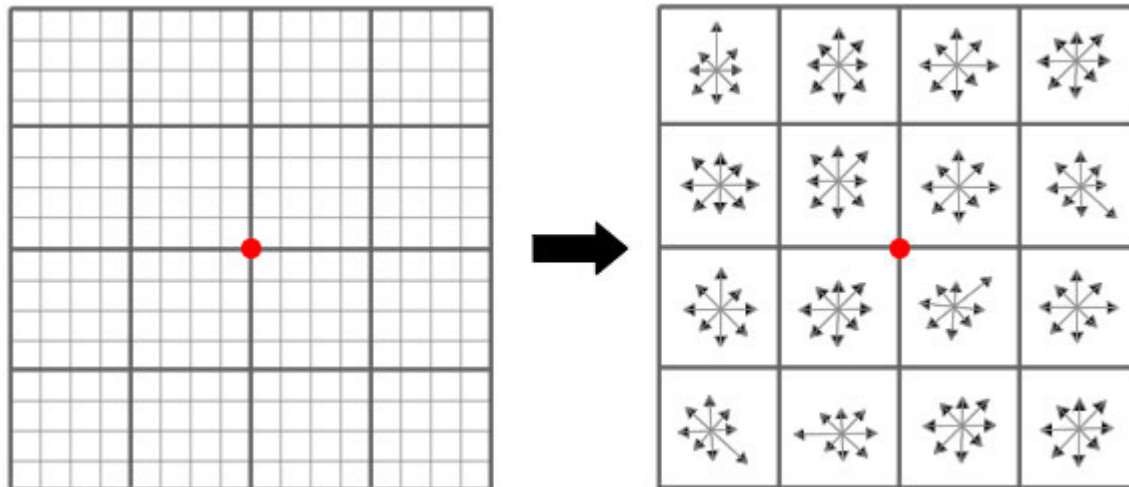


<http://aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/>

SIFT (C7)

5. Crear descriptores

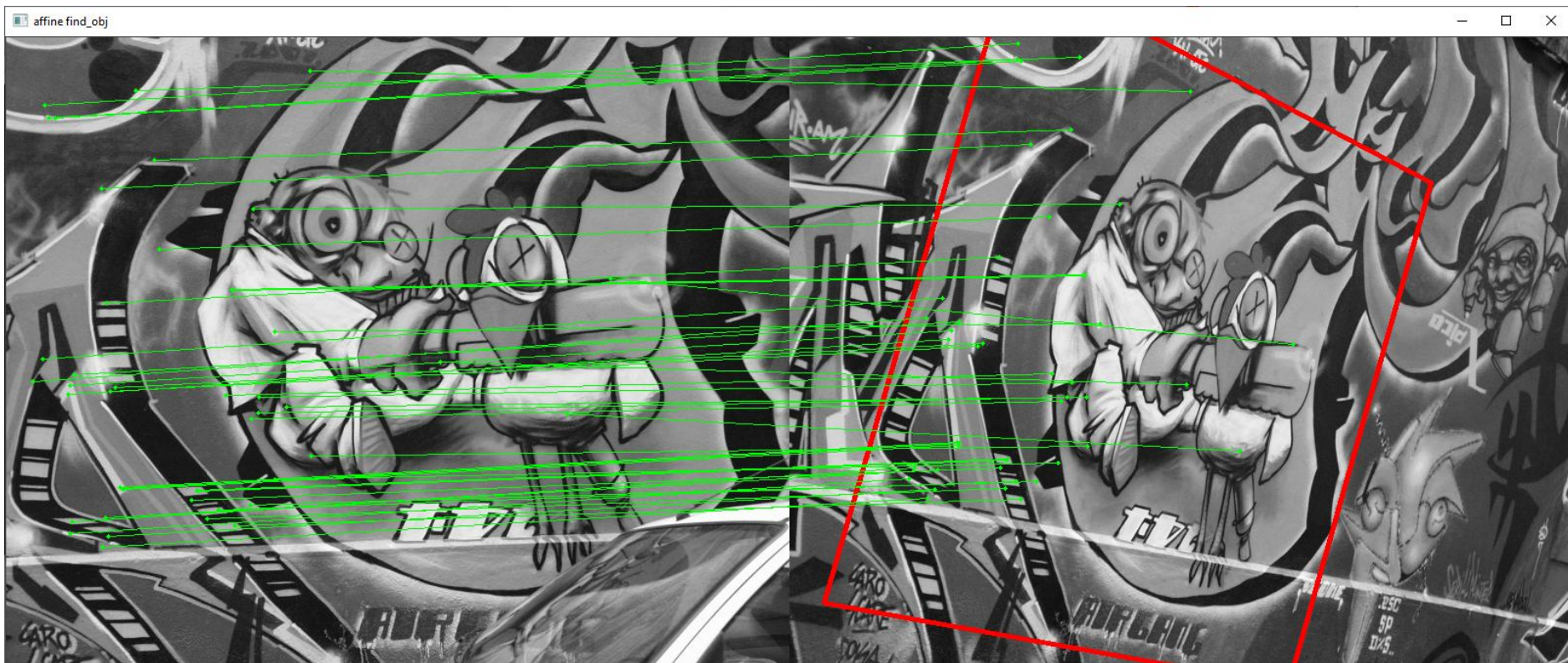
Se centra una ventana de 16×16 en cada punto de interés, luego se divide la ventana en subventanas de 4×4 , en cada subventana se calcula un histograma de 8 bins, se concatenan los valores de los histogramas en un vector de tamaño 128 y finalmente se normaliza el vector para tener magnitud 1



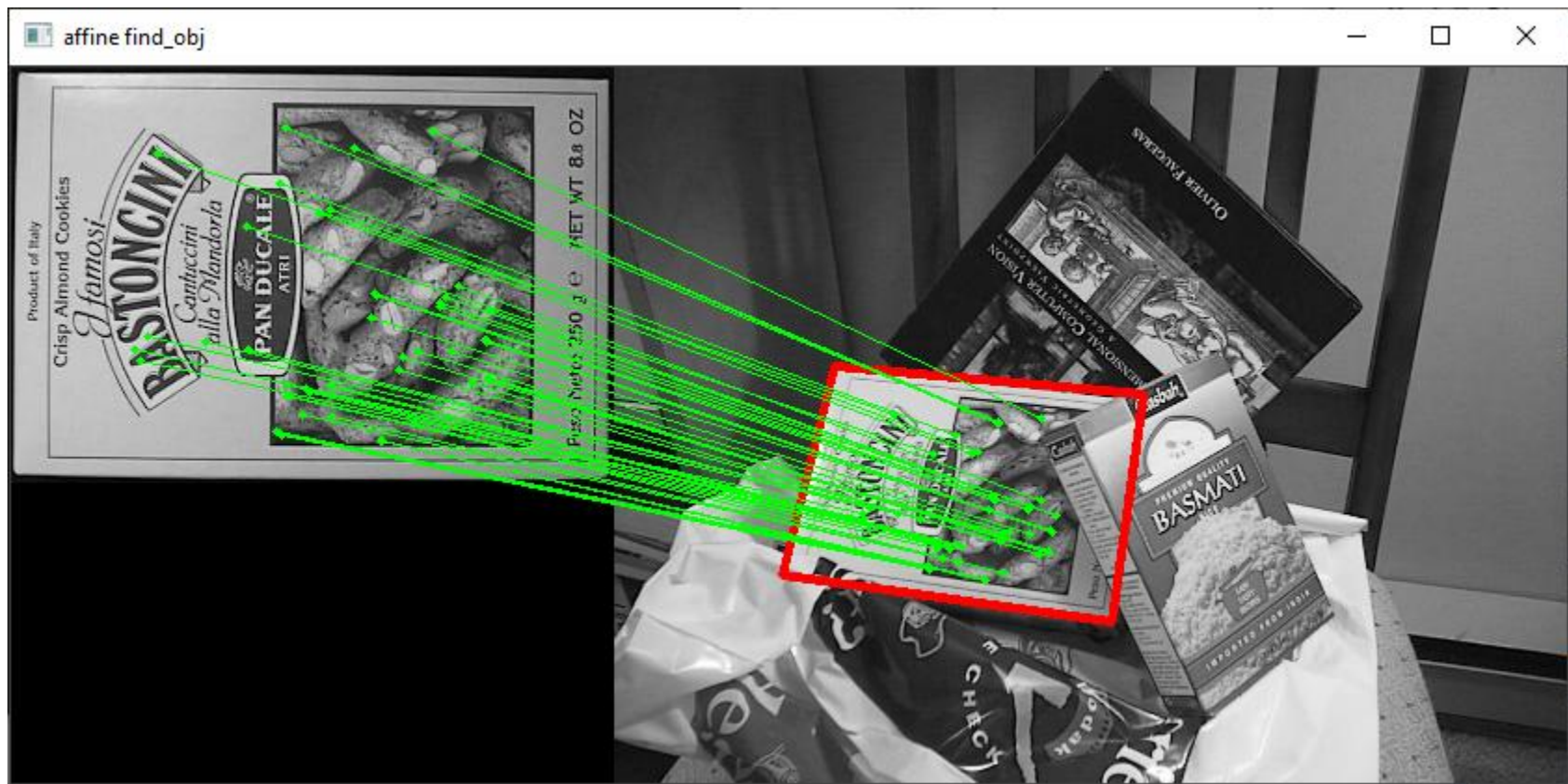
<http://aishack.in/tutorials/sift-scale-invariant-feature-transform-features/>

Usando la escala y la orientación del punto de interés para seleccionar el nivel de la Gaussiana (para suavizar la imagen) y para rotar las coordenadas del descriptor y orientaciones de los gradientes

EJEMPLO DE OPENCV (ASIFT.CPP)



EJEMPLO DE OPENCV (ASIFT.CPP)



SPEEDED UP ROBUST FEATURE (SURF)

- Calidad comparable con la de SIFT pero a un menor costo computacional
- Partes importantes del algoritmo:
 - **Reconstrucción de la imagen.** Utiliza imágenes integrales para reducir el tiempo computacional
 - **Localización de puntos de interés.** Usando la matriz Hessiana
 - **Asignación de orientación.** Usando Haar-wavelet se identifica una orientación reproducible para cada punto de interés
 - **Descriptores.** Se obtienen construyendo una región cuadrada centrada en cada punto de interés y usando las orientaciones previamente calculadas

Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

Avola, D., et al. "Low-level feature detectors and descriptors for smart image and video analysis: A comparative study." *Bridging the semantic gap in image and video analysis*. Springer, Cham, 2018. 7-29.

Puntos de Interés y Descriptores. Francisco J. Hernández-López

Ene-Jun 2025

SURF (C1)

1. Reconstrucción de la imagen usando imágenes integrales

$$ii(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} f(i, j)$$

						0	0	0	0	0	0
1	2	2	4	1		0	1	3	5	9	10
3	4	1	5	2		0	4	10	13	22	25
2	3	3	2	4		0	6	15	21	32	39
4	1	5	4	6		0	10	20	31	46	59
6	3	2	1	3		0	16	29	42	58	74
<i>f</i>						<i>ii</i>					

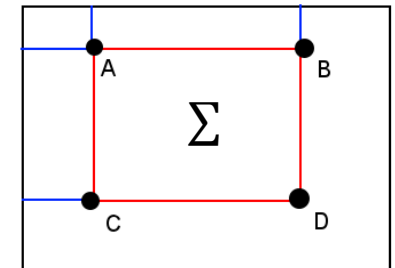
<https://la.mathworks.com/help/images/integral-image.html>

1. Let $s(i, j)$ denote a cumulative row sum, and set $s(i, -1) = 0$.
2. Let $ii(i, j)$ be an integral image, and set $ii(-1, j) = 0$.
3. Make a single row-by-row pass through the image. For each pixel (i, j) calculate the cumulative row sums $s(i, j)$ and the integral image value $ii(i, j)$ using

$$s(i, j) = s(i, j - 1) + f(i, j), \quad (4.2)$$

$$ii(i, j) = ii(i - 1, j) + s(i, j). \quad (4.3)$$

4. After completing a single pass through the image, the integral image ii is constructed.



$$\Sigma = D - B - C + A$$

Algoritmo para construir la imagen integral. Tomado de [Sonka, 2014]

Sonka, Milan, Vaclav Hlavac, and Roger Boyle. Image processing, analysis, and machine vision. Cengage Learning, 2014.

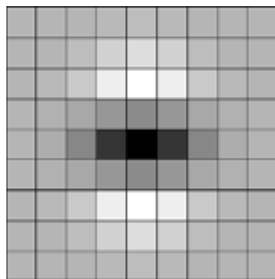
SURF (C2)

2. Localización de puntos de interés usando la matriz Hessiana

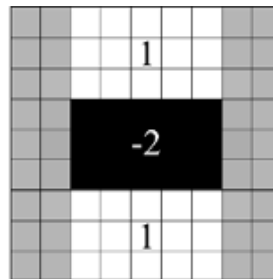
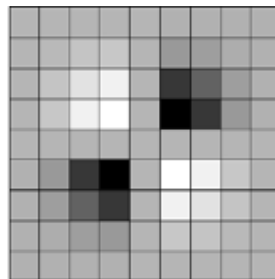
$$H(p, \sigma) = \begin{bmatrix} L_{xx}(\vec{x}, \sigma) & L_{xy}(\vec{x}, \sigma) \\ L_{xy}(\vec{x}, \sigma) & L_{yy}(\vec{x}, \sigma) \end{bmatrix}$$

donde L_{xx} , L_{xy} y L_{yy} son las segundas derivadas aplicando un filtro Gaussiano con parámetro σ en el pixel $\vec{x}$

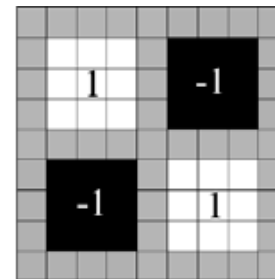
Se utiliza una aprox. a las segundas derivadas usando filtros de caja que pueden ser calculados de forma muy eficiente usando la imagen integral del paso anterior



Derivadas Gaussianas de
2do. orden



Aprox. a las Derivadas
Gaussianas de 2do. orden

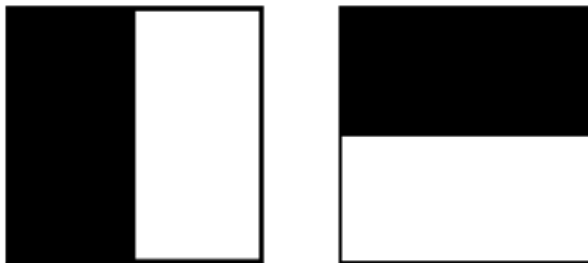


Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

SURF (C3)

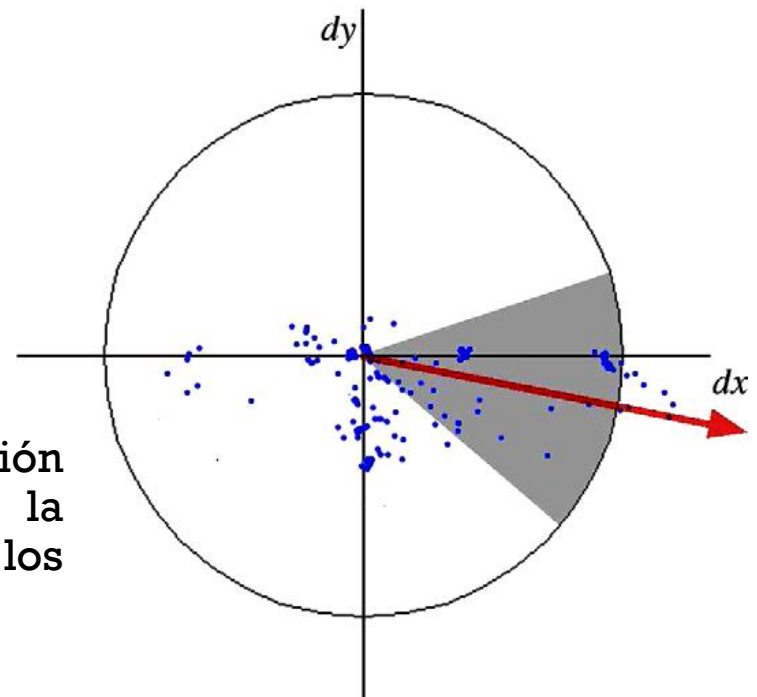
3. Asignación de orientación usando Haar-wavelets

Se aplica un filtro Haar-wavelet en dirección x y y dentro de una región circular de radio $6s$ alrededor del punto de interés, con s la escala en la cual el punto de interés fue detectado. Aquí también se utiliza la imagen integral



Filtros Haar-wavelet

Se recorre una ventana de orientación deslizante de tamaño $\pi/3$ para detectar la orientación dominante de la respuesta de los filtros Haar-wavelet



Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

SURF (C4)

4. Descriptores

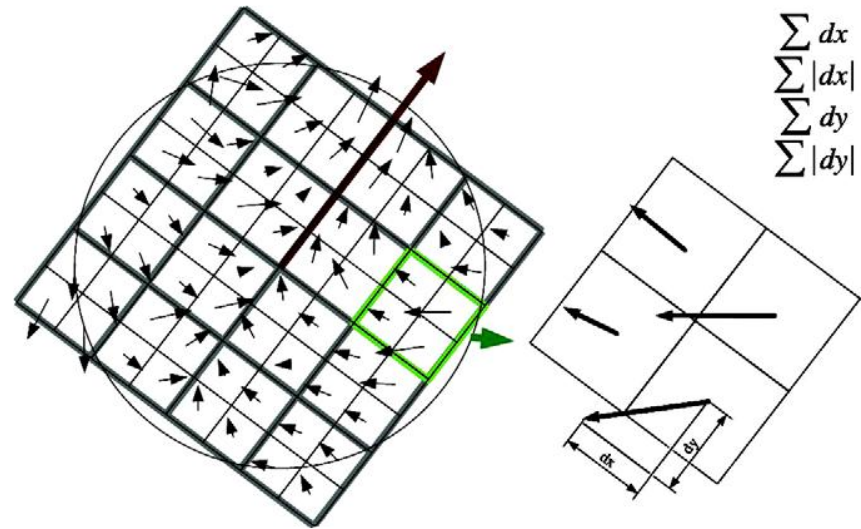
Se obtienen construyendo una región cuadrada centrada en cada punto de interés y usando las orientaciones seleccionadas en el paso anterior. El tamaño de la ventana es $20s$

La region se divide en 4×4 subregiones

Para cada subregión se aplican las Haar-wavelet en una muestra de 5×5 puntos regularmente espaciados, dando como resultado (d_x, d_y)

$d_x \rightarrow$ resp. del filtro en dir. hor.

$d_y \rightarrow$ resp. del filtro en dir. ver.



Se suman las d_x y d_y formando un vector de caract. de la sig. manera: $\vec{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Concatenando estos vectores para las 4×4 subregiones, tenemos un vector de descriptores de tamaño 64

Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

EJEMPLO APLICANDO SURF



Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

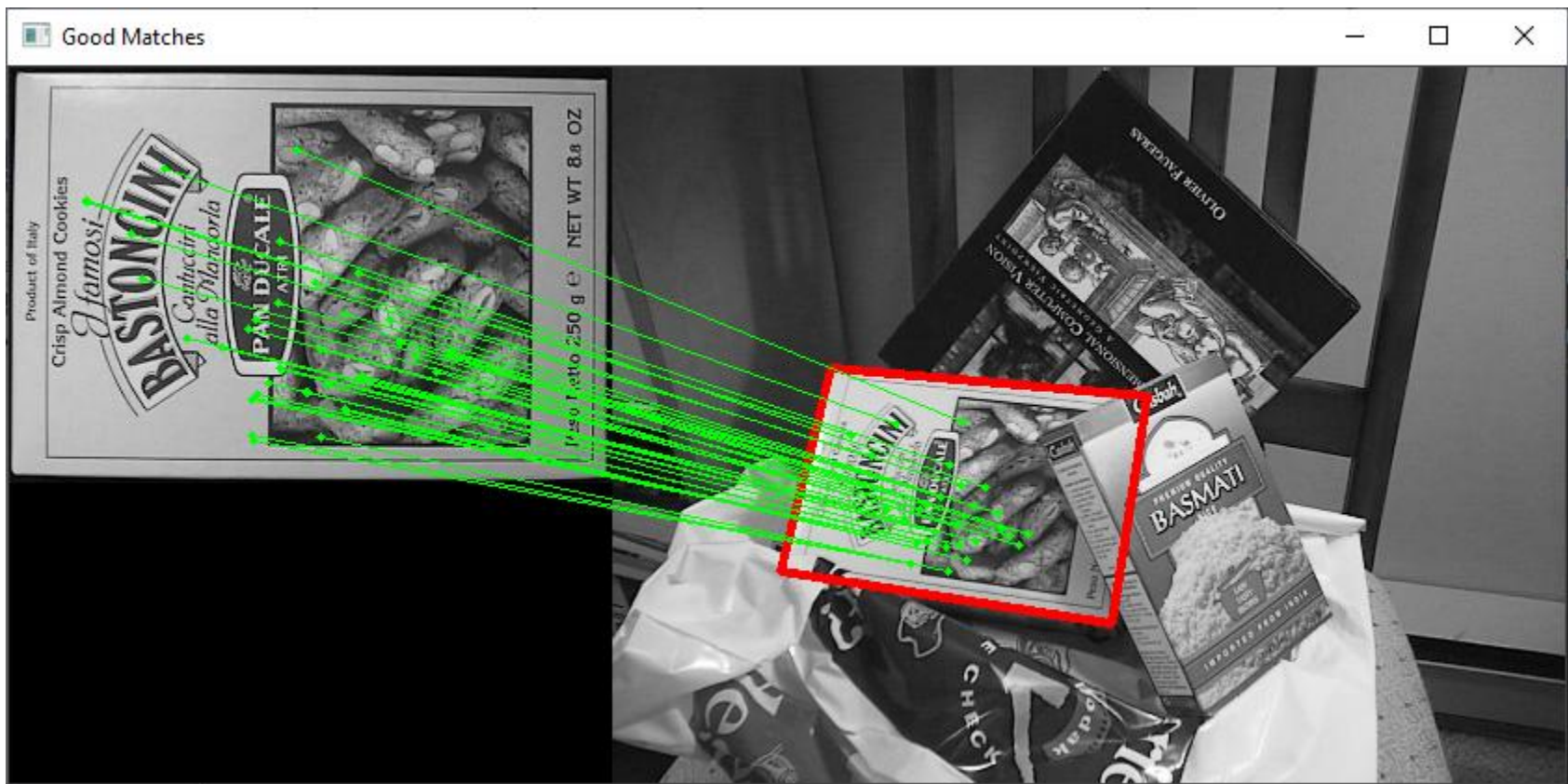
Puntos de Interés y Descriptores. Francisco J. Hernández-López

Ene-Jun 2025

EJEMPLO OPENCV (SURF_FLANN_MATCHING_DEMO)



EJEMPLO OPENCV (SURF_FLANN_MATCHING_DEMO)



GRACIAS POR SU ATENCIÓN

Francisco J. Hernandez-Lopez

fcoj23@cimat.mx

WebPage:

www.cimat.mx/~fcoj23

