

**Universidad del Valle de Guatemala**

Facultad de Ingeniería

## **Laboratorio 3**

**CC3182 – Visión por Computadora**

**Profesor:**

Luis Alberto Suriano Saravia

**Integrantes:**

Diederich Solís — 22952

Andy Fuentes — 22944

Christian Echeverría — 221441

**Fecha: 10/02/2026**

# 1. Task 1: Análisis Teórico del Detector de Harris

En esta sección se desarrolla el análisis teórico del criterio de Harris sin utilizar librerías, con el objetivo de demostrar comprensión conceptual del detector de esquinas basado en el tensor de estructura.

Se proporcionan las siguientes matrices del segundo momento, calculadas en dos píxeles distintos de una imagen:

$$M = \begin{bmatrix} 120 & 5 \\ 5 & 115 \end{bmatrix} \quad M' = \begin{bmatrix} 200 & 10 \\ 10 & 1 \end{bmatrix}$$

A partir de estas matrices se realiza lo siguiente:

1. Cálculo manual de eigenvalores  $(\lambda_1, \lambda_2)$  para  $M$  y  $M'$ .
2. Cálculo de la respuesta de Harris  $R$  para ambas matrices asumiendo  $k = 0,04$ .
3. Clasificación geométrica del píxel: esquina, borde o región plana, justificando con los eigenvalores.

## 1.1. 1. Cálculo manual de Eigenvalores

Para una matriz simétrica  $2 \times 2$ :

$$A = \begin{bmatrix} a & b \\ b & d \end{bmatrix}, \quad \det(A - \lambda I) = 0$$

$$\Rightarrow (a - \lambda)(d - \lambda) - b^2 = 0$$

### 1.1.1. Matriz $M$

$$(120 - \lambda)(115 - \lambda) - 5^2 = 0$$

$$(120 - \lambda)(115 - \lambda) - 25 = 0$$

$$\lambda^2 - 235\lambda + (120 \cdot 115 - 25) = 0$$

$$120 \cdot 115 = 13800 \Rightarrow \lambda^2 - 235\lambda + 13775 = 0$$

Discriminante:

$$\Delta = 235^2 - 4(13775) = 55225 - 55100 = 125$$

$$\sqrt{125} \approx 11,18$$

Por lo tanto:

$$\lambda_{1,2} = \frac{235 \pm 11,18}{2}$$

$$\lambda_1 \approx \frac{246,18}{2} = 123,09 \quad \lambda_2 \approx \frac{223,82}{2} = 111,91$$

### 1.1.2. Matriz $M'$

$$(200 - \lambda)(1 - \lambda) - 10^2 = 0$$

$$(200 - \lambda)(1 - \lambda) - 100 = 0$$

$$200 - 200\lambda - \lambda + \lambda^2 - 100 = 0$$

$$\lambda^2 - 201\lambda + 100 = 0$$

Discriminante:

$$\Delta = 201^2 - 4(100) = 40401 - 400 = 40001$$

$$\sqrt{40001} \approx 200$$

Entonces:

$$\lambda_{1,2} = \frac{201 \pm 200}{2}$$

$$\lambda_1 \approx \frac{401}{2} = 200,5 \quad \lambda_2 \approx \frac{1}{2} = 0,5$$

## 1.2. 2. Respuesta de Harris

La respuesta de Harris se define como:

$$R = \det(M) - k(\text{trace}(M))^2$$

con  $k = 0,04$ .

### 1.2.1. Para $M$

$$\det(M) = 120 \cdot 115 - 5^2 = 13800 - 25 = 13775$$

$$\text{trace}(M) = 120 + 115 = 235$$

$$R = 13775 - 0,04(235^2)$$

$$235^2 = 55225 \Rightarrow 0,04(55225) = 2209$$

$$R = 13775 - 2209 = 11566$$

### 1.2.2. Para $M'$

$$\det(M') = 200 \cdot 1 - 10^2 = 200 - 100 = 100$$

$$\text{trace}(M') = 200 + 1 = 201$$

$$R = 100 - 0,04(201^2)$$

$$201^2 = 40401 \Rightarrow 0,04(40401) = 1616,04$$

$$R = 100 - 1616,04 = -1516,04$$

## 1.3. 3. Clasificación geométrica (Esquina, Borde o Región Plana)

Según la interpretación de los eigenvalores del tensor de estructura:

- **Región plana:**  $\lambda_1$  y  $\lambda_2$  pequeños (poca variación en ambas direcciones).
- **Borde:** un eigenvalor grande y el otro pequeño (variación dominante en una dirección).
- **Esquina:** ambos eigenvalores grandes y comparables (variación fuerte en dos direcciones).

### 1.3.1. Píxel asociado a $M$

Se obtuvo:

$$\lambda_1 \approx 123,09, \quad \lambda_2 \approx 111,91$$

Ambos eigenvalores son grandes y de magnitud similar, indicando cambios fuertes de intensidad en dos direcciones ortogonales. Además,  $R$  es positivo y alto:

$$R \approx 11566 > 0$$

Por lo tanto, este píxel se clasifica como:

Esquina
---------

### 1.3.2. Píxel asociado a $M'$

Se obtuvo:

$$\lambda_1 \approx 200,5, \quad \lambda_2 \approx 0,5$$

Un eigenvalor es muy grande y el otro es muy pequeño, lo cual indica variación fuerte en una sola dirección. Además, la respuesta de Harris fue negativa:

$$R \approx -1516,04 < 0$$

Por lo tanto, este píxel se clasifica como:

Borde
-------

## 2. Task 2: Sistema de Correspondencia (SIFT y ORB)

En esta sección se implementa un sistema completo de correspondencia entre dos fotografías propias de un objeto con textura (caja de cereal *Nesquik Duo*). La primera imagen corresponde a una vista frontal y la segunda a una vista rotada aproximadamente  $45^\circ$  con un cambio de escala evidente. El objetivo es detectar puntos característicos, describirlos y encontrar correspondencias robustas utilizando SIFT y ORB.

### 2.1. Metodología

El pipeline implementado fue:

1. Carga de ambas imágenes en escala de grises.
2. Detección y descripción de keypoints usando:
  - SIFT
  - ORB
3. Matching con **BFMatcher**:
  - SIFT con norma **L2** (Euclidiana).
  - ORB con norma **Hamming**.
4. Filtro con **Lowe's Ratio Test** usando umbral 0,75:

Se acepta un match si  $d_1 < 0,75 d_2$

donde  $d_1$  es la distancia del mejor vecino y  $d_2$  la del segundo mejor vecino.

5. Visualización final dibujando únicamente las correspondencias que pasan el filtro ("buenos matches").

### 2.2. Resultados cuantitativos

La Tabla 1 resume la cantidad de keypoints detectados y la cantidad de matches antes y después del filtro. En este reporte se consideran como *inliers* las correspondencias que sobreviven Lowe's Ratio Test, ya que el requerimiento solicita mostrar solo los "buenos matches" tras el filtrado.

Tabla 1: Resumen de resultados de correspondencia.

Algoritmo	KP Img1	KP Img2	Matches (antes)	Good matches
SIFT	3057	3667	3057	121
ORB	500	500	500	34

### 2.3. Visualización de keypoints

A continuación se muestran los keypoints detectados para cada algoritmo. Se observa que SIFT detecta una mayor cantidad de puntos, mientras que ORB utiliza un límite de 500 keypoints (configuración por defecto), lo cual reduce el costo computacional.

SIFT Keypoints Imagen 1: 3057 detectados



SIFT Keypoints Imagen 2: 3667 detectados



Figura 1: Keypoints detectados con SIFT. Imagen 1: 3057, Imagen 2: 3667.

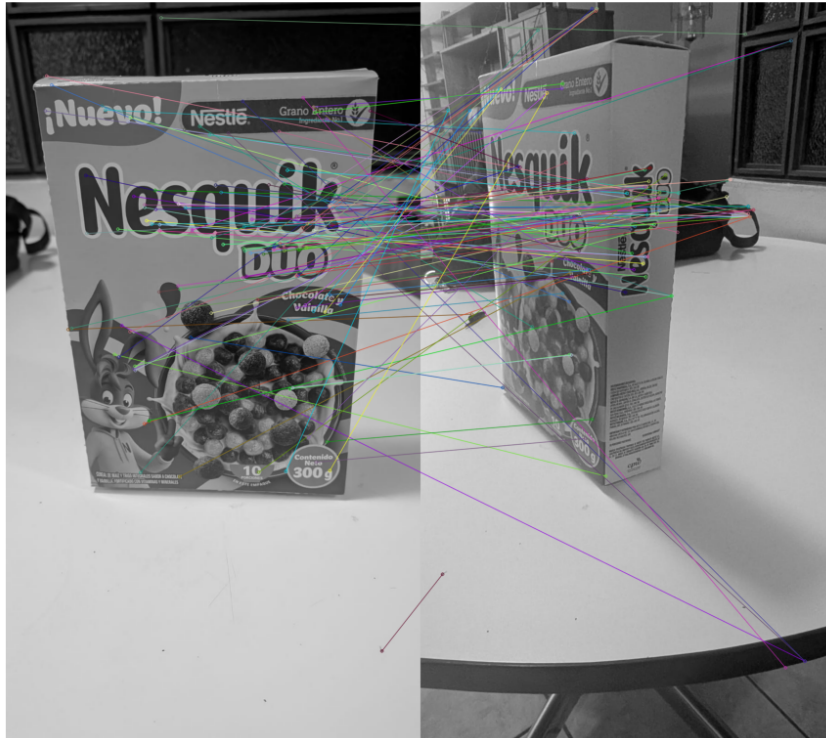


Figura 2: Keypoints detectados con ORB. Imagen 1: 500, Imagen 2: 500.

## 2.4. Correspondencias finales (Good Matches)

Finalmente, se muestran las correspondencias resultantes luego de aplicar Lowe's Ratio Test (0,75). En SIFT se obtuvieron 121 buenos matches, mientras que en ORB se obtuvieron 34 buenos matches.

SIFT: 121 Buenos Matches



ORB: 34 Buenos Matches



Figura 3: Correspondencias finales tras Lowe's Ratio Test (0.75). Arriba: SIFT (121). Abajo: ORB (34).



### 3. Task 3: Evaluación Empírica de Trade-offs (SIFT vs ORB)

En esta sección se mide el costo computacional de los algoritmos implementados en Task 2, separando el tiempo de **detección+descripción** y el tiempo de **matching**. Con base en estos datos y la calidad observada de las correspondencias, se decide qué algoritmo usar en dos productos de drones autónomos.

#### 3.1. Metodología de medición

Para cada algoritmo (SIFT y ORB) se midió el tiempo promedio (en milisegundos) de:

- **Detección + Descripción:** extracción de keypoints y descriptores en ambas imágenes.
- **Matching:** BFMatcher (**L2** para SIFT, **Hamming** para ORB) con `knnMatch(k=2)` y filtrado con **Lowe's Ratio Test** (0,75).

Cada medición se repitió 10 veces y se reporta promedio  $\pm$  desviación estándar.

#### 3.2. Resultados

La Tabla 2 resume los resultados empíricos obtenidos.

Tabla 2: Resultados empíricos de tiempo y correspondencias (promedio  $\pm$  std).

Algoritmo	Det.+Desc. (ms)	Match (ms)	Total (ms)	KP (I1)	KP (I2)	Good
SIFT	219.85 $\pm$ 8.45	21.81 $\pm$ 0.35	241.66	3057	3667	121
ORB	26.11 $\pm$ 1.12	0.55 $\pm$ 0.03	26.66	500	500	34

A partir de estos resultados:

$$\text{FPS estimados} = \frac{1000}{\text{Tiempo Total (ms)}}$$

$$\text{SIFT} \approx \frac{1000}{241,66} = 4,14 \text{ FPS}, \quad \text{ORB} \approx \frac{1000}{26,66} = 37,52 \text{ FPS}$$

Además, ORB resultó aproximadamente:

$$\frac{241,66}{26,66} \approx 9,07 \times \text{ más rápido que SIFT}$$

#### 3.3. Análisis crítico y selección

##### 3.3.1. a. Producto A: Drone de Carreras (Odometría Visual a 60 FPS)

Este producto requiere mínimo 60 FPS, lo cual implica un presupuesto aproximado de:

$$\frac{1000 \text{ ms}}{60} \approx 16,67 \text{ ms}$$

para todo el pipeline.

Con base en la Tabla 2, **ORB** es claramente más rápido que SIFT. Aunque ORB no alcanza los 60 FPS en esta prueba ( $26.66 \text{ ms} \Rightarrow 37.52 \text{ FPS}$ ), es el algoritmo que más se aproxima al requerimiento y el más viable para tiempo real. Para acercarse a 60 FPS en un dron real se requeriría optimización adicional (p. ej., reducir resolución, limitar número de features/ROI, o usar aceleración por hardware).

Elección Producto A: ORB

### 3.3.2. b. Producto B: Drone de Inspección de Grietas (Stitching 4K con precisión alta)

En este caso el tiempo de procesamiento no es crítico, pero la precisión sí. SIFT produjo más keypoints (3057/3667) y más correspondencias válidas (121) bajo rotación y cambio de escala, lo cual es consistente con su robustez ante transformaciones. ORB obtuvo menos matches (34), lo cual reduce redundancia y puede afectar la estabilidad geométrica del stitching cuando hay cambios fuertes de escala/rotación.

Con base en la calidad visual observada, SIFT mantiene más correspondencias útiles para estimar transformaciones con alta precisión, mientras que ORB tiende a perder correspondencias cuando la escala/rotación cambia de forma evidente.

Elección Producto B: SIFT

### 3.3.3. c. ¿Son justas y generalizables estas conclusiones?

Las conclusiones son razonables para este experimento, pero no totalmente generalizables.

#### Limitaciones:

- Solo se evaluó un objeto/escena y un conjunto limitado de condiciones.
- No se varió sistemáticamente iluminación, ruido, desenfoque o *motion blur*.
- No se incluyó verificación geométrica (p. ej., homografía con RANSAC) para contar *inliers* geométricos reales.
- El rendimiento puede cambiar significativamente en hardware embebido del dron.

#### Futuras iteraciones:

- Probar múltiples escenas/texturas y condiciones de iluminación.
- Simular/medir blur y ruido (cámara de baja calidad para Producto A).
- Agregar filtrado geométrico (RANSAC) y reportar *inliers* geométricos.
- Medir latencia en hardware real y evaluar directamente con imágenes 4K para el Producto B.
- Comparar con otros métodos (AKAZE, BRISK) como alternativas intermedias.