

## Taller Parte 1: Estudio Individual del Material de Clase

**Presentado por:**

Juan David Suarez Coronado



Universidad Nacional de Colombia  
Facultad de Ingeniería  
Computación Visual 2025-2S  
Noviembre de 2025

## **1. SIFT (Scale-Invariant Feature Transform)**

### **¿Qué problema resuelve?**

SIFT es un método pensado para reconocer objetos en imágenes aunque cambien mucho entre una foto y otra donde la idea es que el algoritmo pueda identificar lo mismo incluso si el objeto se ve más grande o más pequeño, está girado, la luz es distinta o incluso si una parte está tapada.

### **¿Cuál es su flujo de operación (pipeline)?**

SIFT sigue una serie de pasos para encontrar puntos importantes en la imagen y describirlos de una forma única, casi como una huella dactilar:

1. Primero busca puntos que se mantienen estables cuando se cambia la escala de la imagen y para eso crea versiones más borrosas de la foto y compara una con otra para detectar posibles puntos clave.
2. Luego filtra esos puntos descartando los que tienen poco contraste o los que están en bordes donde es difícil distinguir algo claro.
3. A cada punto clave se le asigna una orientación según la dirección dominante del gradiente en esa zona y esto permite que el punto pueda ser reconocido aunque la imagen está rotada.
4. Al final construye un descriptor donde toma un bloque alrededor del punto, lo divide en partes pequeñas y mide cómo cambian las direcciones de los gradientes y de ahí sale un vector de 128 números que describe ese punto de forma muy precisa.

### **¿Qué ventajas presenta frente a técnicas tradicionales?**

Es increíblemente robusto ya que funciona muy bien incluso cuando las condiciones de la imagen son malas y logra hacer coincidencias confiables aunque el objeto esté muy rotado, a distinta escala o con iluminación complicada y por eso se volvió un estándar durante muchos años.

## **2. ORB (Oriented FAST and Rotated BRIEF)**

### **¿Qué problema resuelve?**

ORB es un método pensado para detectar características en imágenes de forma muy rápida el cual nació porque SIFT y SURF aunque eran muy buenos tenían dos problemas: eran lentos para video en tiempo real y además estaban patentados así que no se podían usar libremente en productos

comerciales, por lo que ORB busca ser rápido, robusto y totalmente libre.

### **¿Cuál es su flujo de operación (pipeline)?**

ORB combina dos métodos conocidos (FAST y BRIEF) y los mejora para que sean invariantes a la rotación y su proceso es así:

1. Primero detecta puntos clave usando FAST que encuentra esquinas comparando el brillo de un píxel con los de un círculo alrededor lo cual hace muy rápido.
2. Luego calcula una orientación para cada punto clave usando el centroide de intensidades de la región cercana lo que permite que el punto pueda reconocerse aunque la imagen esté rotada.
3. Después rota el patrón que usa BRIEF para comparar pares de píxeles y generar el descriptor y así, el descriptor queda alineado con la orientación del punto.
4. Finalmente construye el descriptor BRIEF que es solo una cadena de bits obtenida comparando pares de píxeles (si un píxel es más brillante que otro, se pone 1; si no, 0). Es muy ligero y rápido de comparar.

Para emparejar características entre imágenes, solo compara estos descriptores usando la distancia de Hamming, que básicamente es un XOR entre bits.

### **¿Qué ventajas presenta frente a técnicas tradicionales?**

Porque es rapidísimo, no tiene restricciones de patentes, consume pocos recursos y aun así mantiene una buena robustez a la rotación y a los cambios de escala, por eso es muy usado en robótica, móviles y aplicaciones en tiempo real.

## **3. SLAM (Simultaneous Localization and Mapping)**

### **¿Qué problema resuelve?**

SLAM intenta resolver una pregunta clave para cualquier robot o dron: ¿cómo sabe dónde está y cómo es el lugar por donde se mueve si nunca ha estado allí? y para hacerlo construye un mapa mientras se localiza dentro de él al mismo tiempo y esto es lo que permite que un robot navegue sin GPS o que un celular haga AR de forma precisa.

### **¿Cuál es su flujo de operación (pipeline)?**

En SLAM Visual todo se divide en dos partes: el front-end y el back-end.

#### **Front-End (lo rápido):**

1. Se capturan las imágenes (a veces junto con datos de la IMU).
2. Se buscan puntos clave en las imágenes y se emparejan entre fotogramas consecutivos.
3. Con esas correspondencias se calcula cómo se movió la cámara (rotación y traslación).
4. Con la pose estimada se triangulan los puntos y se arma una nube de puntos 3D básica.

#### **Back-End (lo que corrige y optimiza):**

1. Detecta si el robot vuelve a un lugar ya visitado, lo que ayuda a corregir errores acumulados.
2. Optimiza el mapa y la trayectoria usando métodos como grafo de poses o bundle adjustment.

Al final, SLAM entrega un mapa 3D y la trayectoria exacta de la cámara o robot.

#### **¿Qué ventajas presenta frente a técnicas tradicionales?**

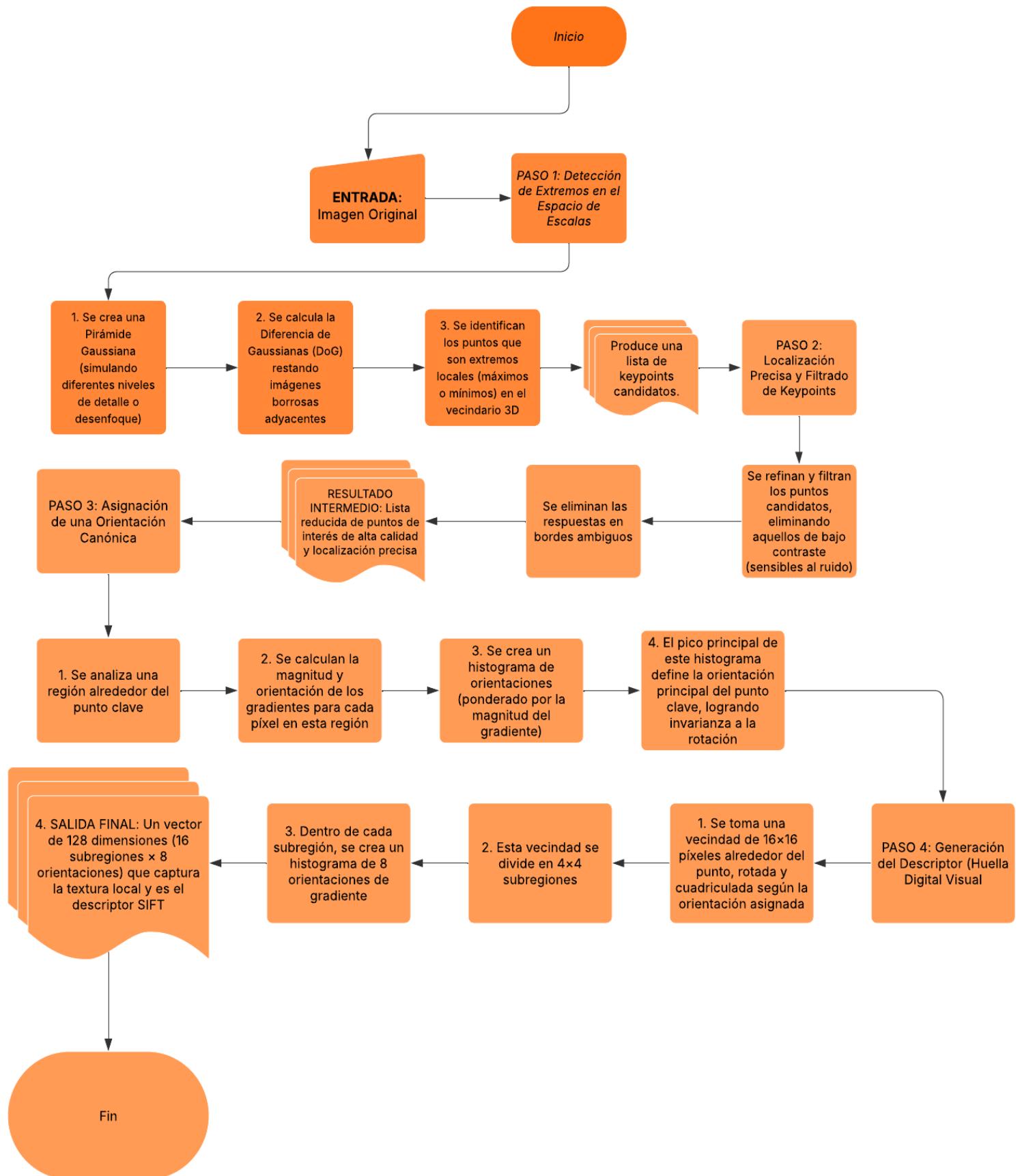
SLAM está hecho para funcionar en tiempo real, mientras que SfM es lento y pensado para procesar las imágenes después.

Además SLAM se enfoca en que el robot sepa dónde está en cada momento y no tanto en obtener una reconstrucción ultra detallada, y gracias al cierre de bucles y la optimización, logra mantener la trayectoria y el mapa consistentes sin que los errores se acumulen.

## 4.1. Diagrama de Flujo del Algoritmo SIFT (Scale-Invariant Feature Transform)

1. Diagrama de Flujo del Algoritmo SIFT (Scale-Invariant Feature Transform)

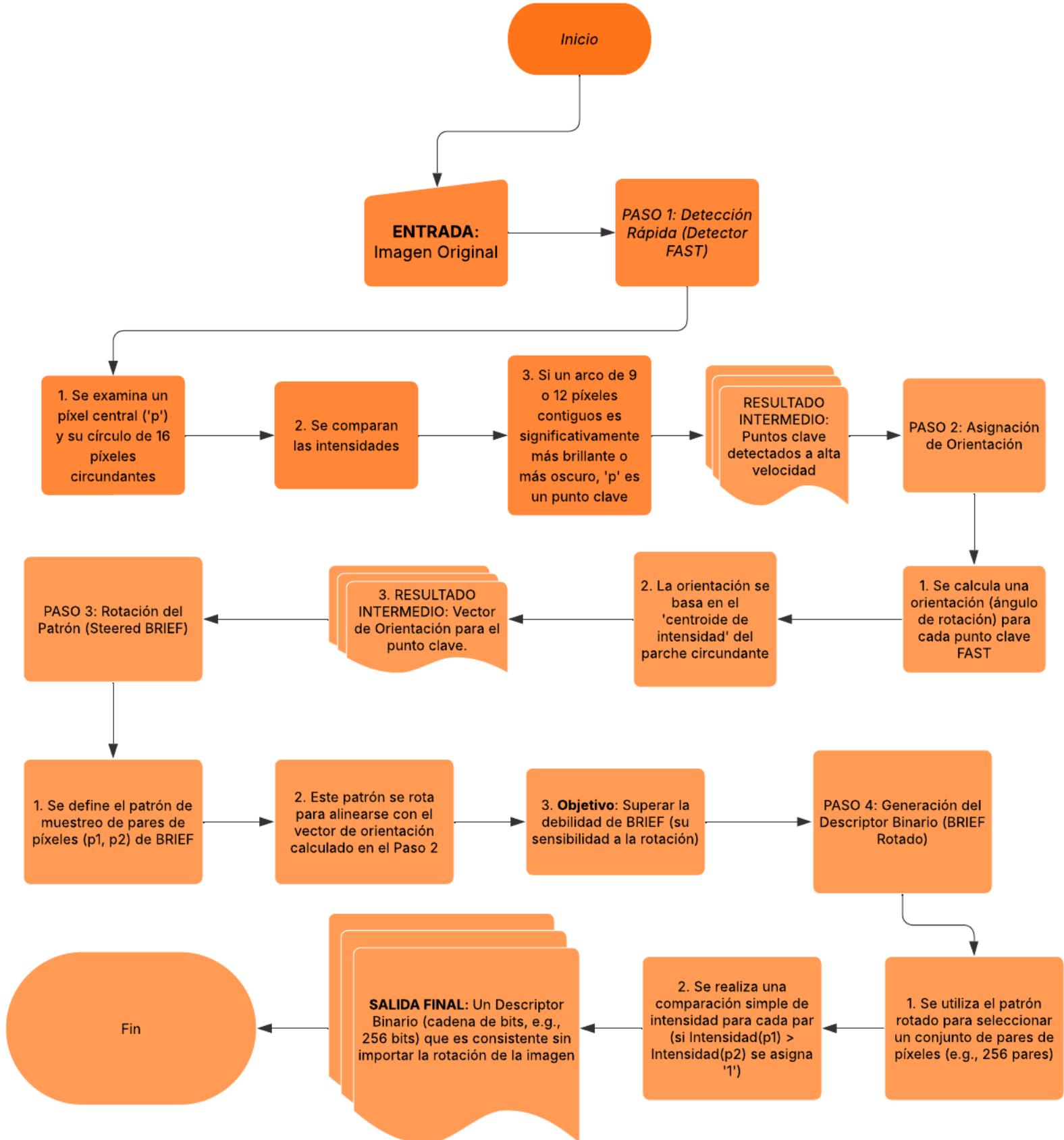
Juan David Suarez Coronado | November 24, 2025



## 4.2. Diagrama de Flujo del Algoritmo ORB (Oriented FAST and Rotated BRIEF)

### 2. Diagrama de Flujo del Algoritmo ORB (Oriented FAST and Rotated BRIEF)

Juan David Suarez Coronado | November 24, 2025



## 4.3. Diagrama de Flujo del SLAM Visual (Simultaneous Localization and Mapping)

### 3. Diagrama de Flujo del SLAM Visual (Simultaneous Localization and Mapping)

Juan David Suarez Coronado | November 24, 2025

