

1. SIFT (Scale-Invariant Feature Transform)

1.1. ¿Qué problema resuelve?

SIFT resuelve el problema de reconocer puntos característicos de un objeto o escena aunque cambien:

- La escala (más grande o más pequeño).
- La rotación (giro del objeto o de la cámara).
- La iluminación.
- La perspectiva (cambios de punto de vista) y cierta oclusión.

En vez de comparar imágenes píxel a píxel, SIFT busca puntos de interés (“keypoints”) y les asigna una especie de “ADN visual”: un descriptor numérico robusto.

Se usa para: reconocimiento de objetos, emparejar imágenes, reconstrucción 3D, etc.

1.2. Flujo de operación (pipeline)

Según la lámina “La Solución de Lowe: SIFT y el ADN Visual” y las diapositivas de pasos 1–4.

1. Construcción del espacio de escalas

- Se genera una pirámide Gaussiana: la misma imagen filtrada con distintos niveles de desenfoque.
- Entre niveles se calcula la Diferencia de Gaussianas (DoG) para resaltar “blobs” o regiones donde la intensidad cambia bruscamente. (Paso 1, jirafa y pirámide Gaussiana).

2. Detección de extremos en el espacio de escalas

- Se buscan máximos y mínimos locales en los volúmenes DoG (comparando un píxel con sus vecinos en la misma escala y escalas adyacentes).
- Estos candidatos son posibles keypoints.

3. Refinamiento y filtrado de keypoints

- Se descartan puntos de bajo contraste (muy sensibles al ruido).
- Se eliminan puntos inestables en bordes usando un criterio tipo detector de esquinas de Harris.
- Resultado: solo quedan keypoints estables y bien localizados. (Gráfica de intensidad fuerte/debil y bordes estable/ambiguo).

4. Asignación de orientación canónica

- Alrededor de cada keypoint se calcula el histograma de orientaciones de gradiente.
- El pico principal de ese histograma define la orientación del punto clave, lo que hace al descriptor invariante a la rotación.
-

5. Construcción del descriptor

- Se toma una ventana alrededor del keypoint, alineada con su orientación.

- Se divide en subregiones (por ejemplo, 4×4). En cada subregión se calcula un pequeño histograma de orientaciones de gradiente.
- Todos esos histogramas se concatenan en un vector de 128 dimensiones (16 subregiones × 8 orientaciones).

6. Emparejamiento (matching)

- Se comparan los vectores de SIFT entre imágenes usando, por ejemplo, distancia Euclidiana.
- Los pares con menor distancia son las correspondencias.

1.3. Ventajas frente a técnicas tradicionales

Comparado con técnicas simples como template matching por correlación directa de píxeles o detectores de bordes sin descriptor:

- Invariante a escala y rotación: puede reconocer el mismo objeto aunque cambie de tamaño o gire.
- Robusto a cambios de iluminación y a cierto ruido.
- Alta precisión en la localización de puntos clave; es considerado un “gold standard” en calidad.
- Sirve como base para tareas más complejas de visión (SLAM, reconstrucción 3D, reconocimiento de objetos en bases de datos grandes).

2. ORB (Oriented FAST and Rotated BRIEF)

2.1. ¿Qué problema resuelve?

ORB se diseñó para mantener buena robustez pero con mucho menor coste computacional que SIFT/SURF y además ser libre de patentes, ideal para tiempo real y sistemas embebidos.

Resuelve:

- Detección y descripción de puntos clave rápida.
- Invarianza razonable a rotación y escala.
- Compatibilidad con dispositivos con recursos limitados (robots móviles, AR en celulares, etc.).

2.2. Flujo de operación (pipeline)

El PDF muestra el procedimiento en pasos: FAST + BRIEF + orientación.

1. Detección rápida de esquinas con FAST

- Para cada píxel ppp , se toma un círculo de 16 píxeles alrededor (radio 3). (Cuadro naranja en el esquema del Paso 1).
- Se compara la intensidad de esos 16 píxeles con la de ppp .
- Si hay un arco de N píxeles consecutivos claramente más brillantes u oscuros que ppp , se marca como punto clave.
- Es extremadamente rápido porque evita operaciones flotantes complejas.

2. Construcción de descriptor binario BRIEF

- Alrededor de cada keypoint se toma un parche de píxeles.
- Se define un conjunto de pares de posiciones $(p_1, p_2), (p_1, p_2), (p_1, p_2)$ dentro del parche.
- Para cada par se compara la intensidad:
 - Si $I(p_1) > I(p_2)$ $I(p_1) > I(p_2)$ $I(p_1) > I(p_2) \rightarrow \text{bit} = 1$
 - Si no $\rightarrow \text{bit} = 0$
- La concatenación de todos esos bits forma un vector binario muy compacto (por ejemplo, 256 bits).

3. Problema de BRIEF: sensibilidad a la rotación

- Si la imagen se rota, los pares $(p_1, p_2), (p_1, p_2), (p_1, p_2)$ ya no representan la misma estructura, y el descriptor cambia completamente.

4. Innovación de ORB: hacer BRIEF “orientado”

- **Cálculo de la orientación del punto clave:**
 - Se estima un vector de orientación usando el centroide de intensidades en la vecindad del punto (momento de primer orden).
- **Rotar el patrón de BRIEF:**
 - Antes de calcular el descriptor, se rota el patrón de pares de puntos para alinearlo con esa orientación.
- **Calcular descriptor rotado:**
 - Con el patrón ya alineado, se genera un descriptor que es consistente bajo rotaciones.

5. Matching

- Como los descriptores son binarios, se usan distancias tipo Hamming (número de bits diferentes), muy rápidas de calcular.

2.3. Ventajas frente a técnicas tradicionales

Comparado con SIFT/SURF y con métodos clásicos:

- Velocidad muy alta: ideal para tiempo real (robots, AR, SLAM visual).
- Descriptor muy compacto: menos memoria, mejor para grandes cantidades de puntos.
- Libre de patentes y con excelente soporte en OpenCV.
- Invariante a rotación y con cierta tolerancia a cambios de escala.
Muy usado dentro de sistemas de SLAM moderno, como ORB-SLAM, ORB-SLAM2 y ORB-SLAM3, que se apoyan en ORB para extraer y emparejar características.

Limitaciones (para mencionar si quieras balancear): menos preciso en texturas muy suaves o con cambios extremos de iluminación.

3. SLAM Visual (Simultaneous Localization and Mapping)

3.1. ¿Qué problema resuelve?

SLAM responde a la pregunta:

¿Cómo sabe una máquina dónde está y cómo es el mundo que la rodea, si no tiene un mapa previo?

Problema central:

- Localización: estimar la pose (posición y orientación) del robot/cámara.
- Mapeo: construir un mapa 3D del entorno al mismo tiempo, usando solo los sensores (cámara, LIDAR, etc.).

Se usa en: vehículos autónomos, robots de servicio, realidad aumentada/mxta, reconstrucción 3D y gemelos digitales.

3.2. Flujo de operación (pipeline general)

Basado en los diagramas de “*El Motor Oculto de la Percepción Espacial*” (proceso de SLAM Visual) y “*SLAM: Enseñando a una máquina a ver, navegar y recordar*” (las dos “mentes”).

Vista de alto nivel

1. **Entrada: captura de datos**
 - Flujo de imágenes de una cámara (SLAM visual monocular) o nube de puntos de LIDAR.
 2. **Front-End (Odometría Visual) – “Mente rápida”**
 - **Extracción de características**
 - Algoritmos como ORB o SIFT detectan puntos clave y calculan descriptores.
 - **Correspondencia de características**
 - Se emparejan keypoints entre fotogramas consecutivos.
 - **Estimación de pose relativa**
 - Con esas correspondencias 2D–2D se usa geometría epipolar (Matriz Esencial/Fundamental) para recuperar la rotación y traslación entre vistas.
 - **Triangulación y construcción del mapa local**
 - A partir de varias vistas se triangulan los puntos en 3D, generando una nube de puntos inicial.
 - Salida del front-end: una trayectoria incremental (odometría visual) propensa a deriva.
3. **Problema: deriva acumulada**
 - Cada estimación de pose tiene un pequeño error; al acumularse, el mapa se deforma y la trayectoria deja de ser precisa.
4. **Back-End – “Mente cartógrafa”**
 - **Detección de cierre de bucle (loop closure)**

- El sistema reconoce que ha vuelto a un lugar ya visitado usando técnicas tipo Bag-of-Words sobre descriptores visuales.
- **Optimización del grafo de poses**
 - Cada pose de la cámara es un nodo; las restricciones (mediciones) son aristas.
 - Al detectar un bucle, se añade una restricción fuerte que corrige la deriva.
- **Bundle Adjustment** (refinamiento final)
 - Ajusta simultáneamente todas las poses y todas las posiciones 3D de puntos para minimizar el error de reproyección.

5. Salida

- Mapa 3D coherente (nube de puntos, malla o mapa semántico).
- Trayectoria precisa del sensor a lo largo del tiempo.

3.3. Ventajas frente a técnicas tradicionales

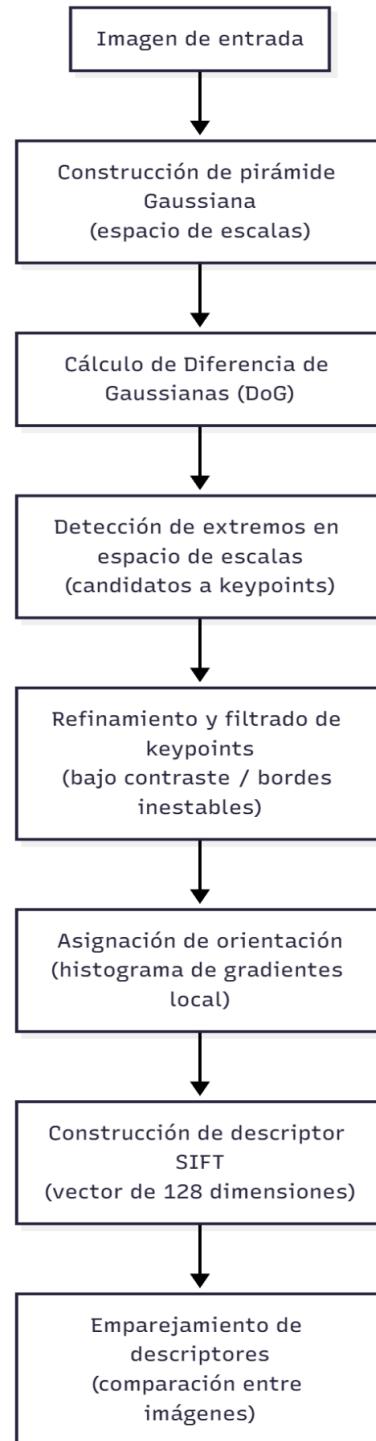
Comparado con aproximaciones clásicas como:

- Solo usar GPS para localización.
- O usar un mapa preconstruido sin capacidad de actualizarlo.

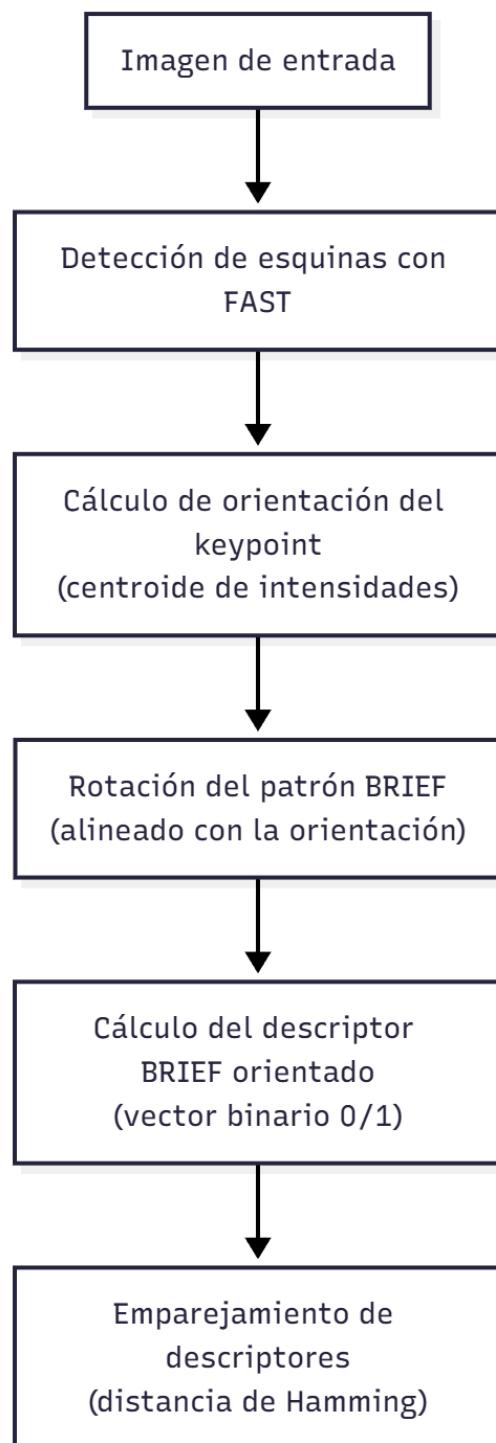
SLAM aporta:

- Autonomía en entornos desconocidos: el robot no necesita mapa previo.
- Actualización continua del mapa: si el entorno cambia, el sistema lo refleja.
- Alta precisión local combinando visión, geometría y optimización.
- Habilita aplicaciones críticas: navegación de vehículos autónomos, robótica industrial, realidad aumentada con tracking 6DoF estable, reconstrucción 3D a gran escala

SIFT (Scale-Invariant Feature Transform)



ORB (Oriented FAST and Rotated BRIEF)



SLAM Visual (Simultaneous Localization and Mapping)

