

Actividad de Profundización: Algoritmos Modernos de Visión por Computador

Angel Humberto Olivera Pinzon

Objetivo General

Comprender el funcionamiento de los algoritmos de visión por computador más relevantes en la actualidad —en particular aquellos utilizados para emparejamiento de características, reconstrucción 3D y modelado neuronal de escenas— mediante análisis individual y discusión colaborativa.

Parte 1: Estudio Individual del Material de Clase

Algoritmo #1: Stitching

¿Qué problema resuelve?

El stitching es la técnica de unir múltiples imágenes para crear una única imagen panorámica o de alta resolución. Se utiliza en fotografía para crear fotos panorámicas de 360 grados o de gran detalle, así como en diagnóstico por imagen para combinar escaneos de áreas extensas en una sola vista completa. También es el algoritmo usado por las cámaras 360° modernas.

¿Cuál es su flujo de operación (pipeline)?



¿Qué ventajas presenta frente a técnicas tradicionales?

1. Mayor campo de visión sin necesidad de equipo especializado

Las técnicas tradicionales requieren:

- **Lentes gran angulares**
- **Ojos de pez**
- **Equipos de escaneo de amplio formato**

El stitching logra **panoramas y vistas de 360° usando cámaras comunes**, reduciendo costos y dependencia de hardware especializado.

2. Mayor resolución que cualquier foto individual

En vez de usar un sensor muy costoso para alta resolución:

- Se capturan varias fotos más pequeñas.
- El algoritmo las combina para crear una imagen **gigapixel o muy detallada**.

Esto supera la limitación física del sensor.

3. Compensación de distorsiones y variaciones

Los algoritmos modernos:

- Corrigen **perspectiva**
- Ajustan **brillo/exposición**
- Reducen **distorsiones de lente**

- Suavizan cambios de color

Las técnicas tradicionales no ofrecen esta corrección automática.

4. Flexibilidad y control sobre la escena

Puedes decidir:

- Qué partes capturar
- Cuántas fotos usar
- Qué nivel de detalle conservar

Mientras que técnicas tradicionales dependen de un único disparo limitado por el lente o la máquina.

5. Menor error humano en áreas grandes

En áreas grandes (médicas, topográficas, industriales), capturar manualmente:

- Requiere mover el equipo
- Puede generar traslapes irregulares
- Depende mucho del operador

El stitching **automatiza la unión y alineación**, reduciendo errores.

6. Ideal para digitalización y documentación

Combina múltiples escaneos para obtener:

- Documentos grandes

- Mapas
- Tejidos biológicos
- Radiografías de zonas extensas

Los métodos tradicionales limitan el tamaño al área de un solo escaneo.

7. Mejor aprovechamiento de datos

Cada foto aporta algo:

- El algoritmo alinea puntos clave
- Aprovecha superposiciones
- Aumenta información espacial

Con métodos tradicionales, cualquier recorte o falta de campo es una pérdida permanente.

8. Mayor precisión en aplicaciones técnicas

El stitching permite:

- Reconstrucciones geométricas precisas
- Ensamblajes exactos de imágenes
- Correcta correspondencia de patrones

Esto es esencial en:

- Diagnóstico médico

- Cartografía
 - Inspección industrial
-

9. Mejora estética

El resultado suele ser:

- Más inmersivo
- Más limpio
- Libre de bordes o solapamientos visibles

Las técnicas tradicionales producen:

- Bordes duros
 - Variaciones de color
 - Distorsiones en un solo frame
-

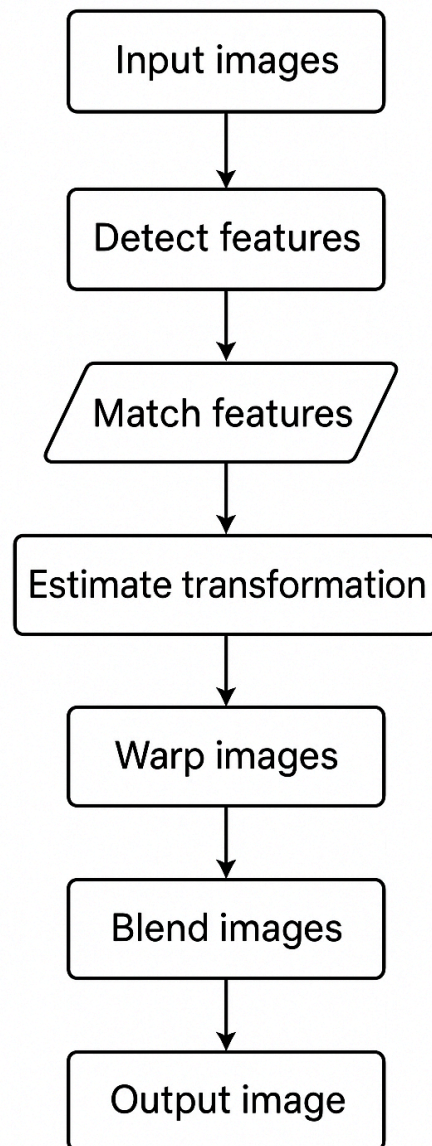
10. Adaptable a inteligencia artificial

Muchos algoritmos modernos usan:

- Detección automática de características (SIFT, SURF, ORB)
- IA para homografía
- Corrección de parallax

Las técnicas tradicionales no pueden aprovechar estas mejoras.

Diagrama de flujo



Algoritmo #2: ORB

ORB fue desarrollado por Ethan Rublee, Vincent Rabaud, Kurt Konolige y Gary R. Bradski en 2011. Combina y mejora dos algoritmos preexistentes:

1. **FAST (Features from Accelerated Segment Test):** Se utiliza para la **detección** eficiente de los puntos clave (corners) en una imagen. FAST es conocido por su **alta velocidad** de cómputo.
2. **BRIEF (Binary Robust Independent Elementary Features):** Se utiliza para la **descripción** de los puntos clave detectados. BRIEF es un descriptor binario que también es muy rápido de calcular y comparar.

ORB toma la velocidad de FAST y de BRIEF, pero añade dos mejoras cruciales que abordan las limitaciones de sus predecesores:

- **Orientación (Oriented):** Asigna una **orientación** a cada punto clave detectado por FAST. Esto se logra calculando el centro de masa del parche alrededor del punto.
- **Rotación (Rotated):** Modifica el descriptor BRIEF para que sea **invariante a la rotación**. Esto significa que el descriptor será el mismo incluso si la imagen está rotada.

¿Qué problema resuelve?

El problema fundamental que resuelve ORB, y los algoritmos de detección y descripción de características en general, es la **identificación y comparación robusta de los mismos puntos o regiones en diferentes imágenes**, a pesar de variaciones como la rotación, el cambio de escala, la iluminación o el ruido.

Específicamente, ORB busca resolver las siguientes limitaciones de sus algoritmos base:

1. **Falta de Invariancia a la Rotación (Limitación de FAST y BRIEF):**
 - **Problema:** Los detectores y descriptores originales como FAST y BRIEF no se desempeñan bien cuando una imagen está rotada, lo que es común en aplicaciones del mundo real.
 - **Solución de ORB:** Al añadir la orientación y rotar los patrones de muestreo del descriptor BRIEF, ORB se vuelve **invariante a la rotación**, permitiendo emparejar características incluso si el objeto o la cámara han rotado.
2. **Lento Tiempo de Cómputo (Limitación de SIFT/SURF):**
 - **Problema:** Algoritmos más antiguos y robustos como SIFT (Scale-Invariant Feature Transform) y SURF (Speeded Up Robust Features) son muy precisos y son invariantes a la escala y rotación, pero son **computacionalmente muy costosos** y lentos para aplicaciones en tiempo real (como robótica móvil o realidad aumentada).

- **Solución de ORB:** ORB es significativamente **más rápido de calcular y emparejar** (al usar un descriptor binario como BRIEF), lo que lo hace ideal para aplicaciones que requieren alta velocidad sin sacrificar demasiada precisión. A menudo se le describe como una alternativa de **alto rendimiento en tiempo real** a SIFT y SURF.

Aplicaciones Comunes

Gracias a su velocidad y robustez, ORB se utiliza en:

- **Stitching de Imágenes (Image Stitching):** Para alinear y fusionar múltiples imágenes en un panorama.
- **Reconocimiento de Objetos (Object Recognition):** Para identificar objetos en diferentes poses.
- **Seguimiento y Mapeo (SLAM - Simultaneous Localization and Mapping):** En robótica y drones para que el dispositivo sepa dónde está y mapee su entorno simultáneamente.
- **Realidad Aumentada (AR):** Para rastrear marcadores o puntos clave en el mundo real para superponer contenido digital.

¿Cuál es su flujo de operación (pipeline)?



¿Qué ventajas presenta frente a técnicas tradicionales?

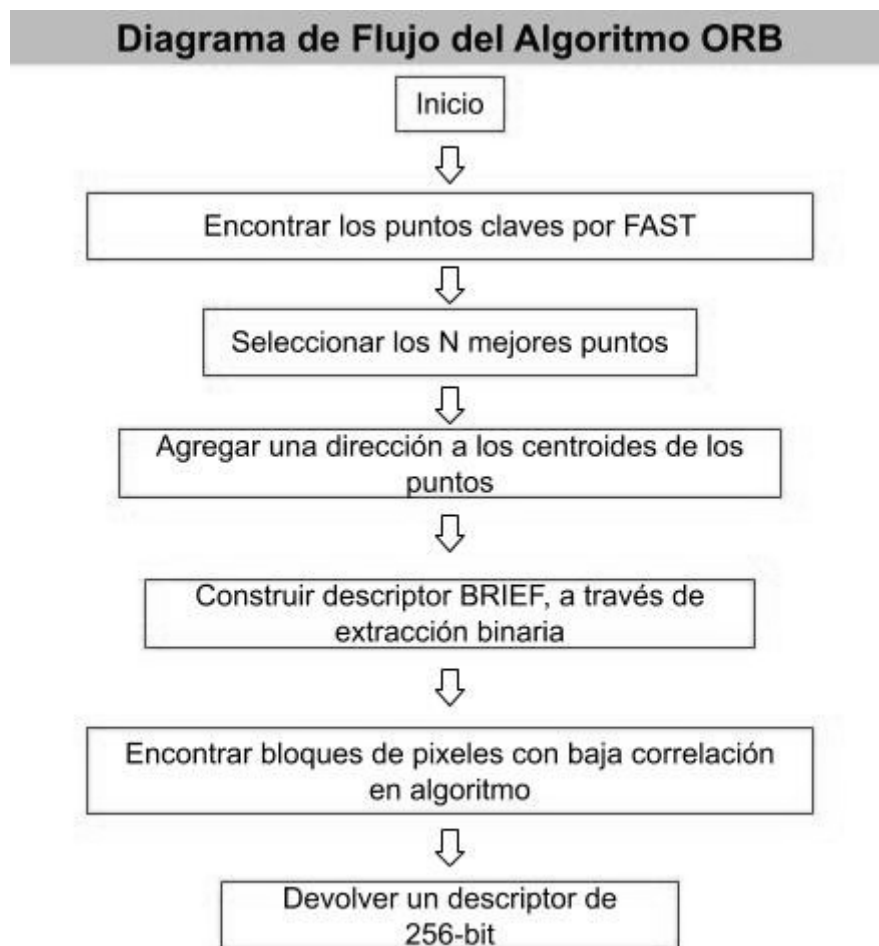
Las técnicas tradicionales de detección y descripción de características incluyen los detectores como **Harris** y **FAST** (sin orientación) y los descriptores como **SIFT** (Scale-Invariant Feature Transform) y **SURF** (Speeded Up Robust Features).

ORB está diseñado específicamente para combinar la robustez de SIFT/SURF con la velocidad de FAST/BRIEF, ofreciendo las siguientes ventajas clave:

Ventaja	Explicación	Frente a SIFT/SURF	Frente a FAST/BRIEF (no orientados)
Alta Velocidad	El uso de FAST para la detección y BRIEF (binario) para la descripción garantiza un tiempo de cómputo extremadamente bajo.	Mucho más rápido para la detección y descripción. Ideal para aplicaciones en tiempo real.	Igual de rápido en detección, pero el descriptor es más rápido de emparejar (distancia de Hamming).
Invariancia a la Rotación	Al añadir la orientación y usar rBRIEF , el descriptor es robusto a la rotación de la imagen.	Comparable en robustez a la rotación.	Superior. Los descriptores BRIEF puros fallan si la imagen está rotada.
Descriptor Compacto	El descriptor binario ocupa menos espacio en memoria y el emparejamiento es muy rápido con la distancia de Hamming.	El descriptor SIFT/SURF es un vector de números de punto flotante de alta dimensión (e.g., 128 o 64 elementos), lo que requiere más memoria y cálculos más lentos.	Superior en robustez (rBRIEF está mejor optimizado para varianza).

Open Source	ORB es de código abierto (publicado bajo la licencia BSD) y no tiene costos de licencia.	SIFT y SURF tenían patentes asociadas (aunque ya expiraron), lo que limitaba su uso en proyectos comerciales sin licencia.	—
--------------------	--	--	---

Diagrama de flujo



Algoritmo #3:

¿Qué problema resuelve?

¿Cuál es su flujo de operación (pipeline)?

¿Qué ventajas presenta frente a técnicas tradicionales?

Diagrama de flujo