

Informe del miniproyecto 4 de Visión Por Computador

**Aplicación de detección de
features y creación de un
panorama**

**Víctor Marcuello Baquero (741278)
Félix Ozcoz Eraso (801108)**

1. Introducción

Las imágenes panorámicas son aquellas que muestran un panorama y que se distinguen por el horizonte visual que cubre. La creación de imágenes panorámicas ha evolucionado a lo largo de la historia, desde la materia pictórica (representación de panoramas mediante cuadros) hasta el diseño de imágenes mediante un dispositivo tecnológico, principalmente, un ordenador.

La aplicación práctica de los panoramas está orientada, principalmente, al diseño computacional gráfico utilizado en disciplinas audiovisuales como la cinematografía y los videojuegos y, por supuesto, también está orientada a la disciplina artística, en concreto, a la fotografía.

En la actualidad, existen múltiples técnicas empleadas para la construcción de panoramas, entre las cuales, se encuentra la composición de un panorama en función de múltiples imágenes que comparten una serie de características en común, características empleadas para la superposición de las mismas y cuyo resultado es una imagen panorámica (a esta técnica se le conoce como stitching).

Los objetivos del proyecto están dirigidos a la introducción en el diseño de panoramas utilizando, como principales parámetros de entrada, una colección de imágenes asociadas a diferentes secciones de una imagen panorámica resultante. En este caso, dichas imágenes se asocian a la fachada de un edificio.

2. Detección de features

Los features o atributos son los elementos visuales presentes en diferentes objetos y/o escenas que están plasmados en una imagen y que resultan potencialmente interesantes en métodos de comparación de dicha imagen con otras imágenes que comparten dichos elementos debido a condiciones como el grado de iluminación y el ángulo de perspectiva. Los tipos de atributos más relevantes son las esquinas (puntos de interés), las figuras (blobs) y las líneas que representan los bordes.

Una vez que se han detectado estos atributos, es posible extraer información relevante asociada a diferentes campos como la forma, el color, la textura, el movimiento y las coordenadas que representan la localización de los features en la imagen mediante un conjunto de coordenadas (una única tupla de coordenadas representa un punto geométrico, mientras que múltiples tuplas representan un contorno).

Las principales cualidades que deben representar los detectores son la velocidad de ejecución, la distribución de carga de recursos computacionales, la precisión de localización y la repetibilidad y robustez de los atributos, que facilita la identificación de características comunes entre los atributos.

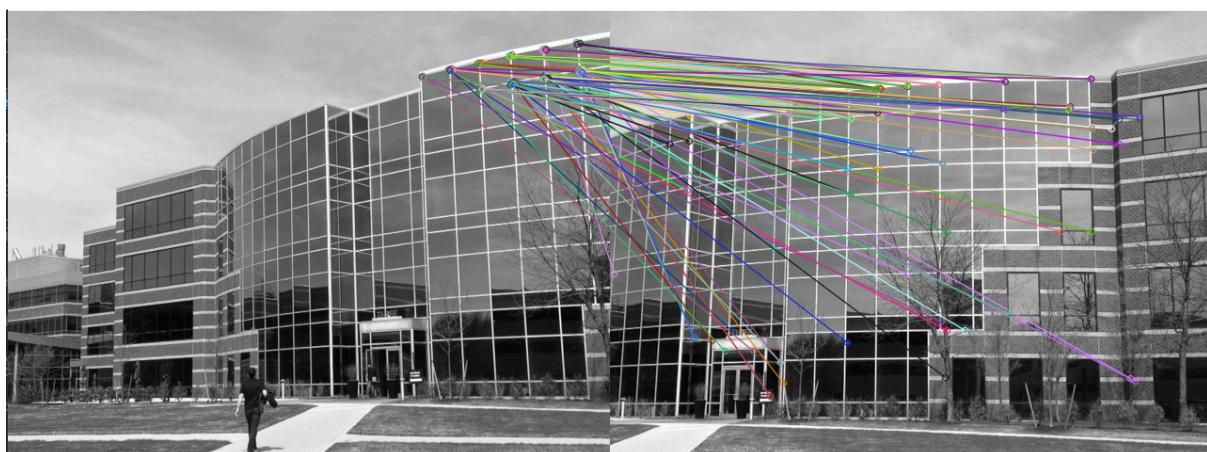
El proyecto incluye una comparación entre los diferentes detectores de atributos que dispone la librería OpenCV, tales como HARRIS (detector de esquinas), ORB, SIFT y AKAZE (el detector SURF resulta bastante farragoso en su instalación, ya que OpenCV no incluye los paquetes asociados al mismo).

Con el propósito de analizar la comparación de atributos detectados entre 2 imágenes, se lleva a cabo la implementación de métodos de emparejamientos de atributos mediante fuerza bruta y FLANN.

Para llevar a cabo un análisis comparativo general de forma adecuada, se implementan aquellos métodos que permiten registrar la información fundamental como el tiempo de ejecución de los métodos de detección de atributos y de emparejamiento de los mismos, el número de atributos detectados en ambas imágenes y el número de emparejamientos realizados en total.

Un requisito importante para mejorar la calidad de detección de atributos es que la escala de color de las imágenes sea una escala de grises, por lo que es necesario modificar dicha escala en caso de que la escala original sea RGB. No obstante, es posible mostrar los atributos y emparejamientos detectados en imágenes en color.

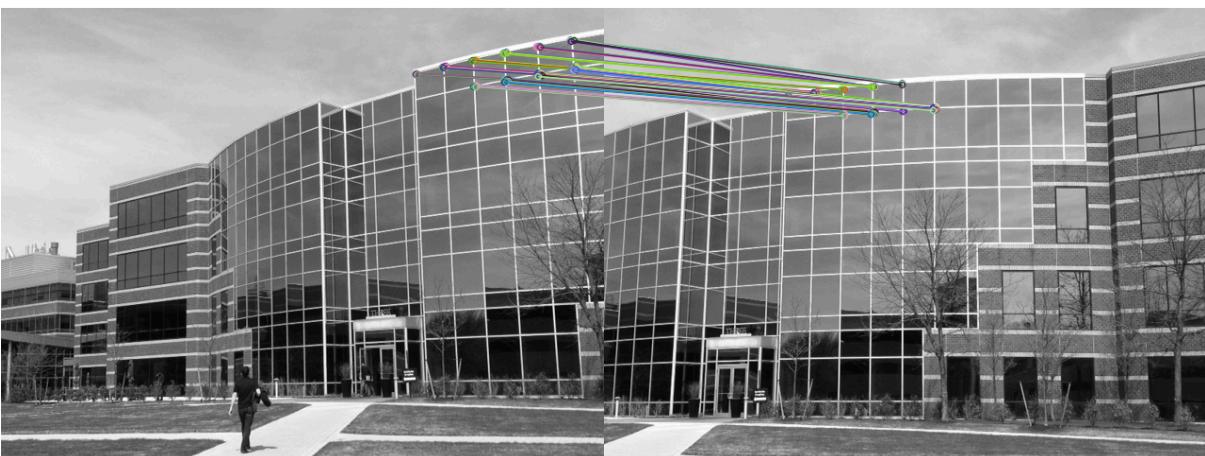
A continuación, se muestran los resultados asociados al análisis comparativo general:





```
Tiempo de ejecuci n de brute_force_matcher1: 1.0830583572387695 segundos
Tiempo de ejecuci n de brute_force_matcher2: 1.1571588516235352 segundos
Tiempo de ejecuci n de harris_matching_function: 2.6141459941864014 segundos
N mero de atributos detectados (imagen 1): 5739
N mero de atributos detectados (imagen 2): 5351
N mero de emparejamientos detectados (emparejador 1): 5259
N mero de emparejamientos detectados (emparejador 2): 5259
```

Figura 1. Emparejamientos, features y m etricas obtenidos para la funci n harris_matching_function cuyo descriptor secundario es ORB, con n_features = 100, n_matches = 100 y con threshold = 0.90





```
Tiempo de ejecuci n de harris_matching_function: 0.03408169746398926 segundos
Tiempo de ejecuci n de brute_force_matcher1: 1.328289270401001 segundos
Tiempo de ejecuci n de brute_force_matcher2: 0.9848647117614746 segundos
Tiempo de ejecuci n de harris_matching_function: 2.4935455322265625 segundos
N mero de atributos detectados (imagen 1): 5739
N mero de atributos detectados (imagen 2): 5351
N mero de emparejamientos detectados (emparejador 1): 5259
N mero de emparejamientos detectados (emparejador 2): 5259
```

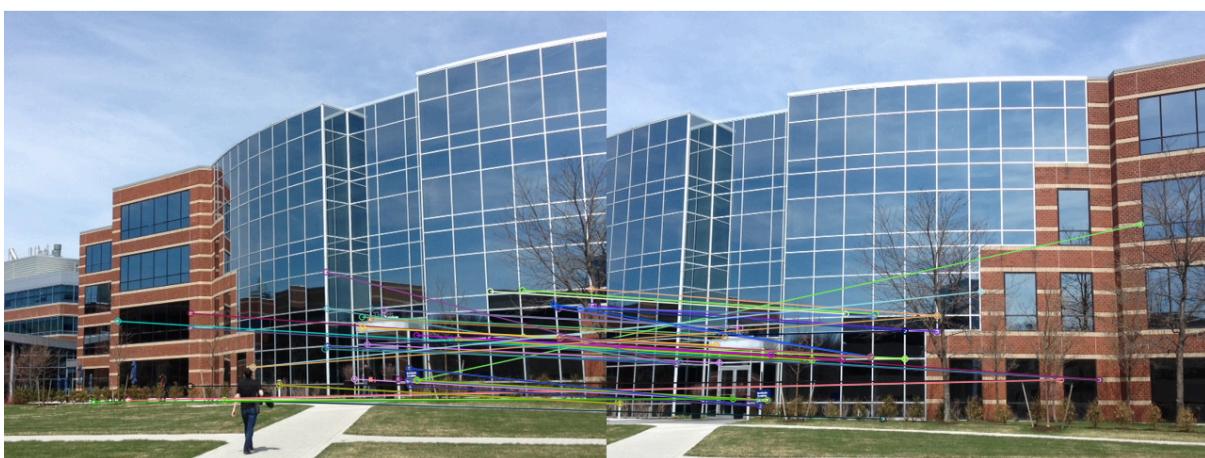
Figura 2. Emparejamientos, features y m etricas obtenidos para la funci n harris_matching_function cuyo descriptor secundario es SIZE y con n_features = 100, n_matches = 100 y con threshold = 0.90





```
Tiempo de ejecuci n de orb_matching_function: 0.8741939067840576 segundos  
Tiempo de ejecuci n de brute_force_matcher1: 0.0 segundos  
Tiempo de ejecuci n de brute_force_matcher2: 0.0 segundos  
Tiempo de ejecuci n de orb_matching_function: 0.07028341293334961 segundos  
N mero de atributos detectados (imagen 1): 100  
N mero de atributos detectados (imagen 2): 100  
N mero de emparejamientos detectados (emparejador 1): 100  
N mero de emparejamientos detectados (emparejador 2): 28
```

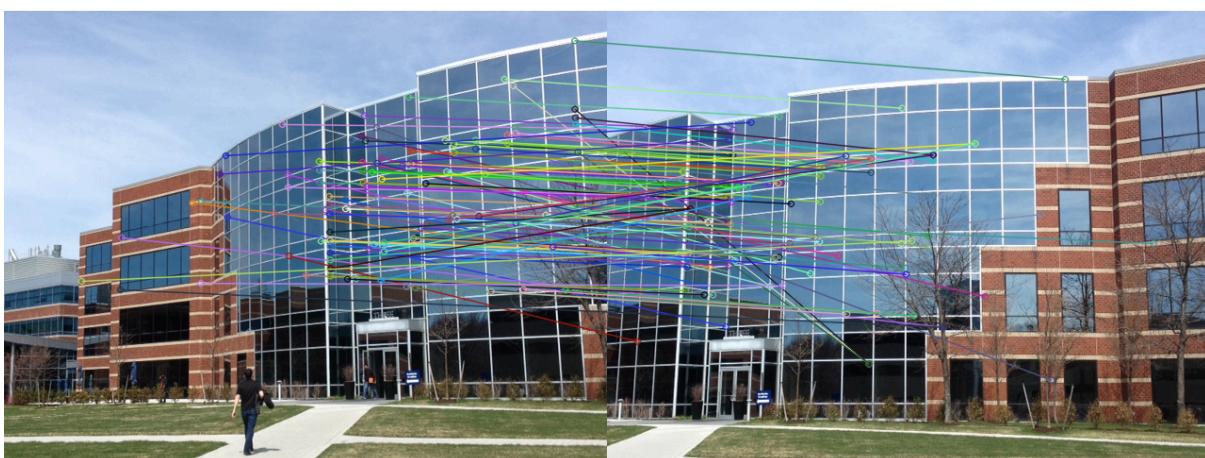
Figura 3. Emparejamientos, features y m etricas obtenidos para la funci n orb_matching_function cuyo descriptor secundario es ORB y con n_features = 100, n_matches = 100 y con threshold = 0.90





```
Tiempo de ejecuci n de harris_matching_function: 0.17597293853759766 segundos  
Tiempo de ejecuci n de brute_force_matcher1: 0.0 segundos  
Tiempo de ejecuci n de brute_force_matcher2: 0.0 segundos  
Tiempo de ejecuci n de sift_matching_function: 0.5652236938476562 segundos  
N mero de atributos detectados (imagen 1): 101  
N mero de atributos detectados (imagen 2): 100  
N mero de emparejamientos detectados (emparejador 1): 101  
N mero de emparejamientos detectados (emparejador 2): 58
```

Figura 4. Emparejamientos, features y m tricas obtenidos para la funci n size_matching_function cuyo descriptor secundario es SIFT y con n_matches = 100 y con threshold = 0.90





Tiempo de ejecución de harris_matching_function: 0.16210222244262695 segundos
Tiempo de ejecución de brute_force_matcher1: 0.22739458084106445 segundos
Tiempo de ejecución de brute_force_matcher2: 0.22699713706970215 segundos
Tiempo de ejecución de akaze_matching_function: 1.1969695091247559 segundos
Número de atributos detectados (imagen 1): 1516
Número de atributos detectados (imagen 2): 1745
Número de emparejamientos detectados (emparejador 1): 1516
Número de emparejamientos detectados (emparejador 2): 675

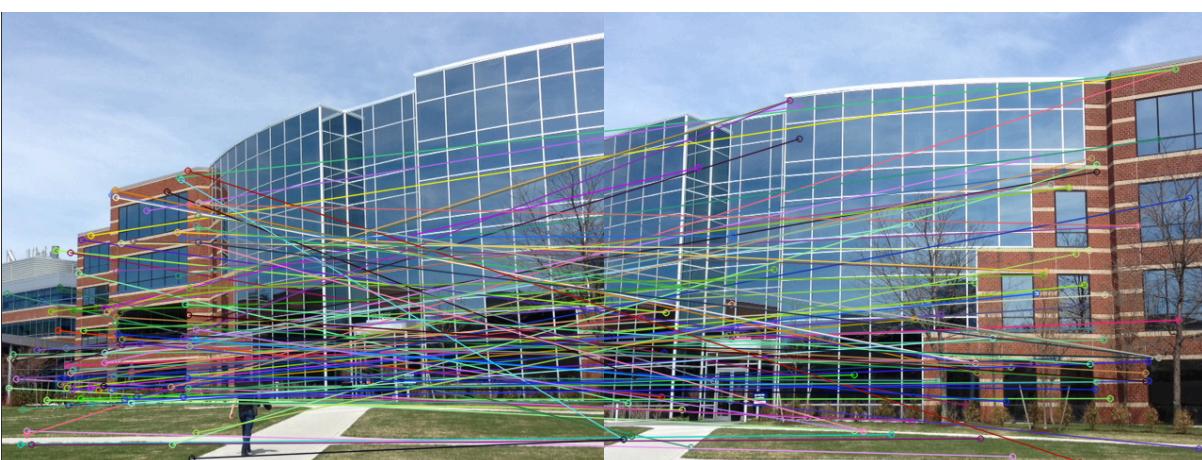
Figura 5. Emparejamientos y métricas obtenidos para la función akaze_matching_function cuyo descriptor secundario es AKAZE y con features = n_100





```
Tiempo de ejecuci n de orb_matching_function: 1.2294981479644775 segundos
Tiempo de ejecuci n de brute_force_matcher1: 1.1643316745758057 segundos
Tiempo de ejecuci n de brute_force_matcher2: 0.9517838954925537 segundos
Tiempo de ejecuci n de orb_matching_function: 2.2488129138946533 segundos
N mero de atributos detectados (imagen 1): 4858
N mero de atributos detectados (imagen 2): 4888
N mero de emparejamientos detectados (emparejador 1): 4858
N mero de emparejamientos detectados (emparejador 2): 1343
```

Figura 6. Emparejamientos, features y m tricas obtenidos para la funci n orb_matching_function cuyo descriptor secundario es ORB y con n_features = 5000, n_matches = 100 y con threshold = 0.90





Tiempo de ejecución de orb_matching_function: 1.2294981479644775 segundos
 Tiempo de ejecución de brute_force_matcher1: 1.1643316745758057 segundos
 Tiempo de ejecución de brute_force_matcher2: 0.9517838954925537 segundos
 Tiempo de ejecución de orb_matching_function: 2.2488129138946533 segundos
 Número de atributos detectados (imagen 1): 4858
 Número de atributos detectados (imagen 2): 4888
 Número de emparejamientos detectados (emparejador 1): 4858
 Número de emparejamientos detectados (emparejador 2): 1343

Figura 7. Emparejamientos, features y métricas obtenidos para la función orb_matching_function cuyo descriptor secundario es SIFT y con n_features = 5000, n_matches = 100 y con threshold = 0.90

Posteriormente, se van a describir los resultados expuestos en las figuras superiores:

- El detector HARRIS se especializa en la detección de esquinas, que representan un punto geométrico de alta relevancia. Para llevar a cabo un emparejamiento, es necesario utilizar un descriptor adicional (se ha utilizado los descriptores ORB y SIZE).
- Los detectores ORB y SIZE disponen del parámetro *n_features*, que permite indicar el número de atributos que se detectan en cada imagen, con el objetivo de optimizar los recursos computacionales empleados para el diseño de panoramas.
- Con el objetivo de comparar los métodos de emparejamiento de fuerza bruta solicitados en el enunciado, se ha comparado los tiempos de

ejecución de ambos métodos. El método de emparejamiento próximo y con cálculo de ratio presenta un tiempo de ejecución menor que el método de emparejamiento próximo simple (*brute_force_matcher1* y *brute_force_matcher2*, correspondientemente).

- El parámetro *n_matches* representa el número de emparejamientos que se muestran en cada par de imágenes. Por esta razón, se muestra el número total de emparejamientos realizados independientemente del valor que presente *n_matches* para ambos métodos de emparejamiento.

3. Creación de un panorama

Partiendo de las conclusiones obtenidas en el estudio comparativo descrito en la sección anterior, se procede a la elaboración de las imágenes panorámicas.

Para realizar las pruebas asociadas a la creación de un panorama a partir de 2 imágenes y cuya composición se realiza en dirección hacia la derecha, se han utilizado las imágenes *BuildingScene*, adjuntas en el enunciado del proyecto. Debido a una mayor precisión de los resultados de emparejamiento obtenidos, se ha optado por utilizar el descriptor SIFT para la creación de imágenes panorámicas.

Debido a la falta de tiempo, no se ha podido mejorar la calidad de los panoramas compuestos, así como la implementación de métodos que permiten crear panoramas en cualquiera de las 4 direcciones y otros métodos opcionales.





Figura 8. Panoramas obtenidos a partir de la composición de 2 imágenes que representan secciones adjuntas de la fachada de un edificio.