Generación de Imágenes Panorámicas mediante la Combinación de Múltiples Vistas de una Misma Escena.

Giacometti, Mateo
Ingeniería en Inteligencia Artificial
Universidad de San Andrés
Buenos Aires, Argentina
mgiacometti@udesa.edu.ar

Piotto, Marcos
Ingeniería en Inteligencia Artificial
Universidad de San Andrés
Buenos Aires, Argentina
mpiotto@udesa.edu.ar

I. Introducción

El presente informe tiene como objetivo exponer de manera clara y precisa la metodología utilizada para la creación de imágenes panorámicas a partir de un conjunto de fotografías que capturan diferentes perspectivas de una misma escena. En este caso particular, se han utilizado tres capturas distintas por cada conjunto de imágenes. A lo largo de las secciones siguientes, se describirá detalladamente cada etapa del proceso, explicando los fundamentos teóricos que respaldan cada técnica empleada así como la implementación de las mismas.

Además, se presentarán los resultados obtenidos en cada etapa de la metodología aplicada a dos conjuntos de imágenes. El primer conjunto compuesto por tres perspectivas del edificio *Mario Hirsch* de la *Universidad de San Andrés*, proporcionadas por la cátedra de la materia *Visión Artificial*. El segundo conjunto, propuesto por los autores de este trabajo, incluye tres perspectivas diferentes del edificio *Familia Galperin-Lebach*, también perteneciente a la misma universidad. Este enfoque permite no solo evaluar el rendimiento del proceso en escenarios controlados, sino también su aplicabilidad en imágenes seleccionadas de manera independiente.

II. Método

II-A. Detección y descripción de características visuales (Features)

En numerosas tareas presentes en el ámbito de la visión por computadora, uno de los primeros y mas importantes pasos a realizar es la **detección y descripción de características visuales locales**. La **detección de características visuales** implica localizar puntos o áreas en una imagen que sean únicas o destacadas, y que puedan ser reconocidas fácilmente en imágenes posteriores. Estos puntos se denominan **keypoints**, los cuales pueden ser **esquinas** (puntos donde la intensidad de la imagen cambia bruscamente en dos direcciones), **bordes** (zonas donde la intensidad de la imagen cambia abruptamente en una sola dirección) o **regiones de textura** (áreas que presentan patrones repetidos).

Una vez que se detectan los puntos clave, es necesario describirlos de una manera que los haga identificables entre diferentes imágenes. Este paso es crucial para establecer correspondencias entre imágenes. El **descriptor** es una representación matemática del entorno local de cada punto clave, generalmente en forma de un vector de características. Este vector captura la información relevante del punto, como la distribución de intensidades de los píxeles en su vecindad.

Los algoritmos mas conocidos para realizar la detección de características visuales son **Harris Corner Detector**, **SIFT**, **SURF** y **ORB**. En particular, para la implementación realizada, se decidió utiliza el algoritmo considerado el **gold standard** de este área, el **SIFT**.

II-A1. SIFT (Scale-Invariant Feature Transform)

Es un algoritmo extremadamente potente y robusto para la detección y descripción de características visuales, el cual fue propuesto por David Lowe en 1999. Este identifica *puntos de interés* en una imagen y genera *descriptores* que describen el entorno local de cada punto. Estos descriptores permiten emparejar características entre diferentes imágenes, siendo robustos a diferencias de escala, rotaciones o diferencia de iluminación.

El algoritmo se compone de **cuatro etapas principales**:

I. Detección de Extremos de Escala (Keypoint Detection)
En esta etapa, SIFT busca identificar puntos clave que sean invariantes a **cambios de escala**. Esto se logra mediante el uso de una **pirámide de escalas** y el cálculo de las **diferencias de Gaussianas** (DoG). SIFT construye una pirámide de la imagen con diferentes niveles de escala. Cada nivel es una versión suavizada (mediante convolución con un filtro gaussiano) y reducida de la imagen original, permitiendo así que los puntos clave sean detectados en diferentes escalas. En cada nivel de la pirámide, SIFT calcula la diferencia entre dos versiones suavizadas consecutivas de la imagen. Este proceso destaca las variaciones en diferentes escalas, lo que ayuda a identificar puntos clave invariantes a cambios de escala. Los puntos clave se identifican como máximos locales en las imágenes de diferencia de Gaussiana (DoG). Estos máximos locales son los puntos donde la intensidad varía

significativamente, lo que los hace buenos candidatos para ser puntos clave.

II. **Eliminación de Puntos No Estables (Keypoint Refinement)**
Una vez detectados los puntos clave, algunos de ellos pueden no ser estables o estar ubicados en bordes no relevantes. Para mejorar la calidad de los puntos clave, SIFT realiza una etapa de refinamiento donde se utiliza un **ajuste cuadrático** para determinar la ubicación exacta del punto clave, eliminando los puntos clave con baja respuesta o que están en bordes débiles, ya que estos puntos no serán fiables en imágenes con **ruido** o **cambios de iluminación**.

III. **Asignación de Orientación (Keypoint Orientation Assignment)**
Para garantizar la **invariación a la rotación**, se asigna una orientación a cada punto clave basada en los gradientes locales de la imagen. Esto permite que los descriptores SIFT sean robustos a las rotaciones de la imagen. Para cada punto clave, se calcula el gradiente de intensidad (dirección y magnitud) en la vecindad del punto. Luego, se genera un **histograma de orientaciones**, donde se agrupan los gradientes en bins de ángulos (e.g., 0°-360°). La orientación dominante del histograma se asigna al punto clave como su dirección principal, lo que permite que el descriptor SIFT se calcule de manera consistente independientemente de la orientación de la imagen.

IV. **Descripción de Características (Keypoint Descriptor)**
En esta etapa, se crea un **descriptor** robusto para cada punto clave detectado. El descriptor es un vector de 128 dimensiones que describe la estructura local alrededor del punto clave, capturando la información de los gradientes en un área local de la imagen.
Para generarlos lo que se hace es tomar una región de 16x16 píxeles alrededor de cada punto clave. Luego, esta región se divide en subregiones de 4x4 píxeles. Para cada subregión, se calculan los histogramas de orientaciones locales de los gradientes, con 8 posibles orientaciones. Finalmente, el descriptor de cada punto clave es un vector de 128 valores (4x4 subregiones con 8 orientaciones cada una), que captura de manera precisa la estructura local de la imagen.
Este descriptor es **invariante a escala, rotación y robusto ante cambios de iluminación**, lo que lo hace muy adecuado para emparejar puntos clave entre diferentes imágenes.

II-A2. Procedimiento Realizado

Utilizando la función `cv2.SIFT_create()` de la librería **OpenCV** inicializamos un objeto **SIFT** con los mejores parámetros conseguidos. Posteriormente, mediante el método `.detectAndCompute()` del objeto creado, calculamos los puntos clave y descriptores de cada una de las imágenes que compondrán nuestra panorámica.

II-B. Distribución espacial de características visuales (Suppression)

Generalmente, cuando se utilizan algoritmos de detección de características visuales, suelen obtenerse **demasiados puntos clave** en áreas con muchas texturas o bordes, mientras que en otras áreas de la imagen puede haber pocos puntos. Esto genera una distribución no homogénea de los puntos clave, por lo que para obtener una representación más equilibrada de los puntos clave en toda la imagen se utilizan algoritmos para seleccionar estos puntos de forma mas equitativa. En particular, para este trabajo fue utilizado el de Supresión de **No Máxima Adaptativa** o **ANMS** por sus siglas en ingles.

II-B1. ANMS (Adaptive Non-Maximal Suppression)

El algoritmo **ANMS** se utiliza para seleccionar **keypoints** de manera más eficiente y equitativa en una imagen, eliminando aquellos que son redundantes o están muy cercanos entre sí, conservando solo los más representativos.

El proceso comienza con la detección de un conjunto de puntos clave mediante un algoritmo como SIFT. Cada punto clave viene acompañado de una **respuesta**, que indica su relevancia o fuerza en la imagen. Estos puntos suelen concentrarse en zonas con alto contraste o variaciones de textura, lo que puede llevar a redundancias en ciertas áreas. ANMS corrige esta acumulación, distribuyendo los puntos de manera más uniforme.

Para cada punto clave, ANMS calcula lo que se llama el **radio de supresión**. Este radio es la distancia mínima a otro punto con una respuesta mayor. Cuanto más alejado esté un punto clave de otro con mayor respuesta, mayor será su radio de supresión. Esto refleja que un punto clave es más importante si no tiene competidores cercanos más fuertes. Por el contrario, puntos cercanos a otros más relevantes son considerados redundantes.

Una vez calculados los radios de supresión para todos los puntos clave, se ordenan de mayor a menor. Los puntos con los radios más grandes son seleccionados, asegurando que queden distribuidos de manera uniforme en toda la imagen. Esto evita la concentración de puntos en áreas pequeñas, garantizando una cobertura más representativa de las características importantes de la imagen sin redundancia innecesaria.

II-B2. Procedimiento Realizado

Desarrollamos una implementación propia del algoritmo **ANMS**, al que se le proporcionaron todos los puntos clave junto con sus respectivas respuestas y descriptores de cada imagen con el fin de lograr una mejor distribución de estos para cada una de ellas.

II-C. Asociación de características visuales (Matching)

El **matching de características visuales** es un proceso que permite encontrar correspondencias entre puntos clave detectados en diferentes imágenes. El objetivo del técnica es encontrar pares de características que correspondan al mismo punto en diferentes imágenes de la misma escena, a pesar

de posibles cambios en la perspectiva, escala, iluminación u oclusión.

En lo referido a la tarea de crear imágenes panorámicas, lo que se busca hacer mediante el matching es encontrar asociaciones de correspondencias entre las 3 imágenes (en particular, de la imagen central o ancla con respecto a las imágenes de los extremos). Para ello, dentro de las múltiples algoritmos existentes para la resolución de esta tarea, utilizaremos **KNN Matching** para calcular las asociaciones de correspondencias junto al **Lowe Ratio Test** para filtrar las coincidencias erróneas.

II-C1. KNN Matching

Es un algoritmo utilizado para emparejar descriptores de características entre dos imágenes en el contexto de la detección y correspondencia visual. Su funcionamiento se basa en comparar los descriptores de los puntos clave de ambas imágenes y encontrar los K vecinos más cercanos para cada descriptor, permitiendo así identificar las mejores coincidencias entre características visuales.

El funcionamiento de **KNN Matching** consiste en calcular, para cada descriptor en el primer conjunto, su distancia a todos los descriptores en el segundo conjunto utilizando una métrica de distancia adecuada (como L2 para SIFT o Hamming para BRIEF y ORB). El algoritmo luego selecciona los **K descriptores más cercanos** (o coincidencias) para cada descriptor del primer conjunto. Esto significa que, para cada descriptor en la primera imagen, se obtiene una lista de las K mejores coincidencias en la segunda imagen, ordenadas según su proximidad.

II-C2. Lowe Ratio Test

El **Lowe Ratio Test** es una técnica empleada para mejorar la precisión en el emparejamiento de descriptores, filtrando coincidencias incorrectas y reduciendo el número de falsos positivos.

El método funciona comparando la distancia entre la mejor coincidencia y la segunda mejor coincidencia para cada descriptor. Si la relación entre ambas distancias es menor que un umbral predefinido (comúnmente 0.75), se considera que la coincidencia es confiable. En cambio, si la diferencia entre estas distancias no es lo suficientemente significativa, la coincidencia se descarta, ya que podría indicar un falso positivo. Esto permite seleccionar solo las coincidencias más sólidas y confiables.

II-C3. Procedimiento Realizado

Utilizando la función `cv2.BFMatcher()` de **OpenCV** inicializamos un objeto **Brute-Force Matcher**. Posteriormente, mediante el método `.knnMatch()` del objeto con el parámetro **k=2**, aplicamos **KNN Matching** sobre los descriptores para luego pasar los resultados por una implementación propia de **Lowe Ratio Test**, obteniendo así las correspondencias filtradas.

II-D. Estimación de homografía de forma manual

La **homografía** es una transformación proyectiva que relaciona dos imágenes del mismo plano, permitiendo mapear puntos de una imagen a otra cuando ambas capturan una superficie plana desde distintas perspectivas. Es una matriz de transformación 3x3 que describe cómo los puntos de una imagen se corresponden con los de otra bajo una transformación proyectiva.

La **transformación homográfica** se puede representar como:

$$\alpha \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Donde (x, y) son las coordenadas de un punto en la primera imagen, (x', y') las coordenadas del punto correspondiente en la segunda imagen, H es la matriz de homografía (3x3) y α es un factor de escala.

Para cumplir el objetivo del trabajo, requerimos obtener las homografías que logre alinear las 2 imágenes exteriores con la imagen elegida como ancla de la panorámica. A fin de poder estimar las homografias requeridas, usamos el algoritmo **Direct Linear Transformation** o **DLT** por sus siglas.

II-D1. Direct Linear Transformation (DLT)

Es un algoritmo utilizado para estimar la matriz de homografía (o, en general, matrices de proyección) en el contexto de la visión por computadora.

Para poder estimar la homografía, es necesario contar con al menos 4 pares de puntos correspondientes entre las dos imágenes, los cuales pueden ser obtenidos siguiendo los pasos planteados en las secciones anteriores.

Dado un par de puntos correspondientes (x, y) y (x', y'), podemos escribir el sistema de ecuaciones que representa la relación entre ellos en términos de la matriz de homografía H. Expandiendo la fórmula de la homografía:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad , y' \quad = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Donde cada h_{fc} es un elementos correspondiente a la matriz homográfica H.

Podemos reorganizar estas ecuaciones en una forma lineal. Multiplicamos por el denominador, ordenamos un poco los términos y obtenemos las siguientes dos ecuaciones para cada par de puntos:

$$x \cdot h_{11} + y \cdot h_{12} + 1 \cdot h_{13} - x' \cdot (h_{31}x + h_{32}y + h_{33}) = 0$$

$$x \cdot h_{21} + y \cdot h_{22} + 1 \cdot h_{23} - y' \cdot (h_{31}x + h_{32}y + h_{33}) = 0$$

Esto representa dos ecuaciones lineales por cada par de puntos, lo que nos lleva a un sistema lineal de ecuaciones. Dado que hay 8 incógnitas (los elementos de H, sin contar el factor de escala), se necesitan al menos 4 pares de puntos para resolver el sistema.

Si tenemos N correspondencias de puntos, podemos escribir el sistema de ecuaciones en forma matricial como $A \cdot h = 0$, donde A es una matriz de tamaño $2N \times 9$ que contiene los coeficientes de las ecuaciones, y h es un vector columna de los elementos de la matriz de homografía H.

Este sistema se puede resolver utilizando el método de los valores y vectores singulares (SVD), que es una técnica estándar para resolver sistemas de ecuaciones lineales sobredeterminado. El vector h que minimiza el error es el correspondiente al valor singular más pequeño de A. A partir de h, se reconstruye la matriz de homografía H.

II-D2. Procedimiento Realizado

Se establecieron cuatro correspondencias entre la imagen ancla y las dos imágenes periféricas. Estas correspondencias se utilizaron para calcular una homografía mediante una implementación propia del algoritmo DLT. Posteriormente, se compararon los resultados obtenidos con los generados por la función `cv2.findHomography()`, concluyendo que la implementación de DLT es capaz de producir homografías de alta calidad.

II-E. Eliminación de outliers y estimación de homografías de forma algorítmica

Para conseguir la estimación de homografías robustas, es fundamental eliminar o reducir la influencia de puntos mal emparejados o que no corresponden a la misma escena, los cuales pueden generar distorsiones significativas en la proyección de las imágenes y afectar la precisión del modelado de las panorámicas. Estos *outliers* pueden alterar la alineación y causar que la homografía resultante no represente correctamente la relación geométrica entre las imágenes.

Para mitigar su influencia, es necesario emplear un algoritmo que permita detectar y excluir estos puntos fuera de lugar. En este trabajo, se ha elegido utilizar el algoritmo **RANdom SAmple Consensus** o **RANSAC** por sus siglas.

II-E1. RANdom SAmple Consensus (RANSAC)

Es un algoritmo iterativo diseñado para estimar parámetros de un modelo matemático a partir de un conjunto de datos que contiene *outliers*. Su principal objetivo es separar los datos en *inliers* (puntos que siguen el modelo) y *outliers* (puntos que no lo hacen) de manera robusta.

El algoritmo, en el contexto de eliminación de outliers para la construcción de homografías, consiste en seleccionar aleatoriamente cuatro pares de puntos correspondientes entre dos imágenes, el número mínimo necesario para calcular una homografía. A partir de estos cuatro puntos, se calcula una homografía provisional utilizando el método DLT. Luego, esta homografía se aplica al resto de los puntos del conjunto de correspondencias, midiendo la distancia entre la posición transformada de cada punto y su correspondiente en la segunda imagen. Si esta distancia está por debajo de un umbral predefinido, el punto se clasifica como un inlier; de lo contrario, se considera un outlier.

Este proceso de selección aleatoria de puntos, cálculo de la homografía y evaluación de inliers se repite muchas veces a lo largo de un número predeterminado de iteraciones. En cada iteración, se podría encontrar una homografía más adecuada, es decir, la que logra maximizar el número de *inliers*. Al finalizar las iteraciones, se elige la homografía que mejor ha alineado los inliers, considerada como la mejor aproximación a la relación geométrica real entre las imágenes. Finalmente, con los inliers identificados, se recalcula la homografía utilizando únicamente estos puntos válidos, lo que da lugar a un modelo más preciso y robusto.

II-E2. Procedimiento Realizado

Se implementó una versión propia del algoritmo **RANSAC**, a través del cual se filtraron las correspondencias *outliers* previamente generadas. Este proceso permitió obtener matrices de homografía robustas, que serán utilizadas en la siguiente etapa de la metodología.

II-F. Juntando y mezclando imágenes (Stitching and Blending)

II-F1. Cálculo del tamaño de la imagen final

Para determinar el tamaño final del panorama, primero se debe seleccionar una imagen de referencia, denominada imagen ancla. Posteriormente, se calculan las matrices homográficas de las imágenes adyacentes con respecto a la imagen ancla. Estas matrices permiten describir la transformación geométrica necesaria para alinear cada imagen con la de referencia.

Una vez calculadas las matrices homográficas, hay que transformar las esquinas de cada imagen para obtener los límites de la imagen panorámica final, identificando las coordenadas mínimas y máximas del conjunto.

II-F2. Definición del nuevo sistema de coordenadas

El nuevo sistema de coordenadas se define tomando el punto $(-min_x, -min_y)$ como el nuevo origen del lienzo o *canvas*. Esto significa que las matrices homográficas deben ajustarse sumando una traslación de $-min_x$ en la coordenada x y de $-min_y$ en la coordenada y. Alternativamente, es posible recalcular las matrices homográficas en relación con el nuevo origen, con la imagen de referencia trasladada a $(-min_x, -min_y)$.

II-F3. Proyección de las imágenes laterales

Con las matrices homográficas ajustadas, se utilizó la función `cv2.warpPerspective()` de **OpenCV** para proyectar las imágenes laterales en el plano de la imagen ancla. Esto permitió alinear cada imagen lateral con la imagen central.

II-F4. Proceso de blending para suavizar los bordes

Al superponer las imágenes proyectadas, se obtiene una primera aproximación de la imagen panorámica. Sin embargo, aunque las imágenes están correctamente alineadas, los bordes de cada una siguen siendo visibles, lo que afecta la cohesión visual.

Para eliminar los bordes visibles entre las imágenes y lograr una transición suave, se empleó un proceso de blending. Este método consiste en mezclar gradualmente las imágenes en las zonas de superposición, evitando así transiciones bruscas entre las diferentes partes de la panorámica.

El primer paso para realizar el blending es crear una máscara binaria para cada imagen y cada canal de color (RGB). Estas máscaras binarias indican las áreas donde las imágenes tienen información válida (valor 1) y donde no la tienen (valor 0). Luego, utilizando la función `cv2.distanceTransform()` de **OpenCV**, se calcula una matriz de distancias, la cual asigna a cada píxel un valor que representa su distancia al píxel más cercano con valor cero en la máscara.

II-F5. Imagen Final

La imagen final se calculó como una suma ponderada de cada imagen por su correspondiente máscara. Finalmente, se normalizó el resultado dividiendo por la suma de las máscaras, lo que garantiza que no haya diferencias de brillo entre las áreas solapadas.

III. Resultados

La imágenes provistas por la cátedra para la realización de este trabajo son las siguientes:

Figura 1. Fotografías del edificio *Mario Hirsch* de la *UdeSA*

III-A. Detección y descripción de características visuales (Features)

A continuación, se muestra una imagen en la que se encuentran marcados sus puntos de interés generados mediante el uso de **SIFT**:

Figura 2. Keypoints generados por *SIFT* sobre la imagen *ancla* del edificio *Mario Hirsch*

Los puntos de interés mostrados en la imagen son 3000, los cuales fueron tomados de los mas de 102246 generados.

III-B. Distribución espacial de características visuales (Suppression)

Se muestra la misma imagen del inceso anterior posterior a la aplicación de **ANMS**:

Figura 3. Keypoints restantes de la aplicación de *ANMS* sobre la imagen *ancla* del edificio *Mario Hirsch*

Los puntos de interés mostrados en la imagen pasan de 3000 a tan solo 1000.

III-C. Asociación de características visuales (Matching)

Se muestran los matches obtenidos luego de aplicar **KNN Matching** y **Lowe Ratio Test**:

Figura 4. Correspondencias obtenidas de la aplicación de *KNN Matching* y *Lowe Ratio Test* sobre la imagen *ancla* y la imagen derecha del edificio *Mario Hirsch*

Como se puede observar, se reduce el número de keypoints utilizados, quedando únicamente aquellos que tienen una buena correspondencia con puntos en la otra imagen.

III-D. Eliminación de outliersy estimación de homografías

Observamos el resultado de la aplicación de **RANSAC**:

Figura 5. Correspondencias obtenidas de la aplicación *RANSAC* sobre la imagen *ancla* y la imagen derecha del edificio *Mario Hirsch*

Se reduce en cierta medida el número de correspondencias que se utilizarán para calcular la homografía más robusta entre ambas imágenes.

III-E. Juntando y mezclando imágenes (Stitching and Blending)

El siguiente gráfico muestra el resultado de trasladar las esquinas de las imágenes del edificio *Mario Hirsch* de la *Udesa*:

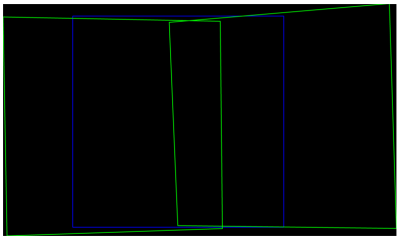

Figura 6. Esquinas transformadas de las imágenes: la imagen ancla en azul y las imágenes trasladadas en verde

Una vez obtenidas las dimensiones de la imagen final y las correspondientes matrices homograficas, podemos proyectar de imágenes las imágenes, tal como se muestra a continuación:

Figura 7. Proyecciones de las imágenes en el plano de la imagen central (Udesa1.jpg)

A continuación se muestra la aproximación inicial superponiendo las imágenes proyectadas:

Figura 8. Imagen panorámica preliminar con bordes visibles

Como se explico en la sección de metodología, para difuminar los bordes de las imágenes es necesario hacer un *blending*. A continuación se muestra un ejemplo de las máscaras generadas:

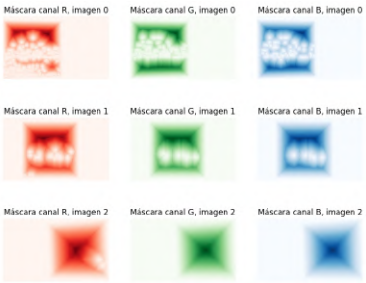

Figura 9. Máscaras de cada imagen del edificio *Mario Hirsch*, separadas por canal RGB

El resultado final del proceso de **blending** se muestra a continuación. Como se puede observar, los bordes entre las imágenes ya no son visibles, y el panorama se percibe como una única imagen continua:

Figura 10. Panorámica final del campus de la Universidad de San Andrés

III-F. Aplicación a conjunto de imágenes propio

Para comprobar que nuestro método es generalizable, aplicamos las mismas técnicas a un conjunto nuevo de imágenes. En particular, el edificio *Galperin-Lebach* de *Udesa*. La imágenes obtenidas fueron la siguientes:

Figura 11. Imágenes del edificio *Galperin-Lebach* de la *UdeSA*

A continuación, se observa el resultado final de la aplicación de la metodología anteriormente expuesta:

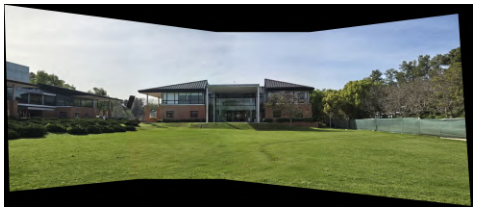

Figura 12. Imagen panorámica del *Galperin-Lebach*

Al igual que en el caso base, aunque los edificios están alineados, se observan defectos en el césped de las imágenes finales. Esto se debe a que los keypoints detectados corresponden solo a los edificios, ignorando la posición del césped.