

Trabajo Práctico 1 - UdeSA Panorámica

1. Introducción:

El propósito de este proyecto es unir imágenes de una misma escena de manera de crear una imagen panorámica “sin costuras”, es decir, que no se noten los puntos de empalme. Cada imagen debe tener un **buen grado de superposición**, es decir, una suficiente cantidad de características visuales locales presentes en ambas imágenes (aproximadamente 30-70% de la imagen).

Un fotógrafo profesional ha venido a la universidad a tomar imágenes pero..
¡Justo ese día estaba lloviendo! y su imagen panorámica quedó algo gris.



¡Es nuestra tarea ayudarlo a realizar una nueva imagen panorámica!

2. Conjuntos de imágenes:

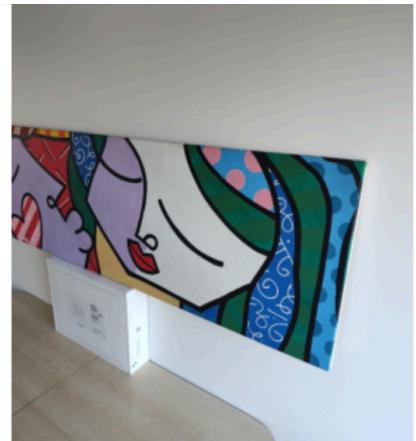
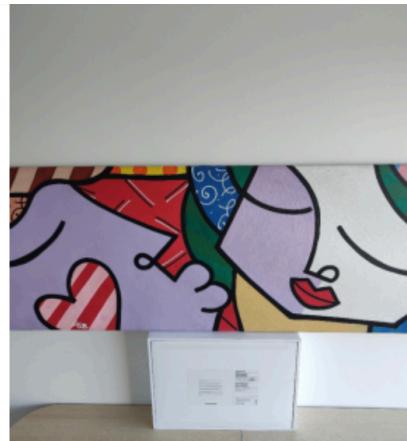
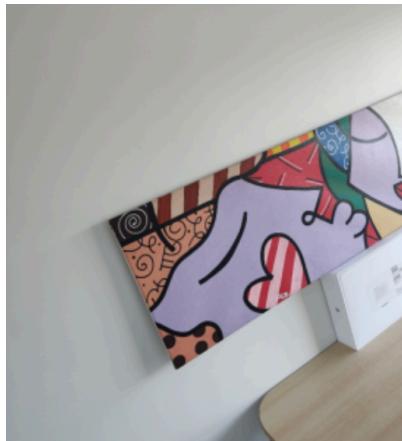
Se proveen dos conjuntos de tres imágenes para que puedan trabajar los diferentes métodos requeridos en el enunciado. Al final del trabajo se les pedirá tomar sus propias imágenes para probar su método, **se sugiere tomarle fotos a la universidad para resolver una nueva imagen panorámica en la que no esté lloviendo.**

Es necesario que la secuencia de imágenes tenga **al menos tres imágenes** con aproximadamente 30 - 70% de superposición. De esas tres imágenes, una de ellas se la considerará la imagen “ancla”, a la que las otras dos deberán ajustarse.

Además, **es importante que las características visuales en las imágenes, utilizadas para calcular las transformaciones homográficas, “pertenezcan a un mismo plano”** de manera que exista una única transformación homográfica dominante entre las imágenes.

En caso de ser un escenario de una estructura compleja (como un edificio, no un objeto plano), **la estructura deberá estar “lo suficientemente lejos”** para que las características visuales puedan considerarse en un “plano en el infinito” y el método funcione correctamente.

Ejemplo de objeto plano con características visuales en un mismo plano:

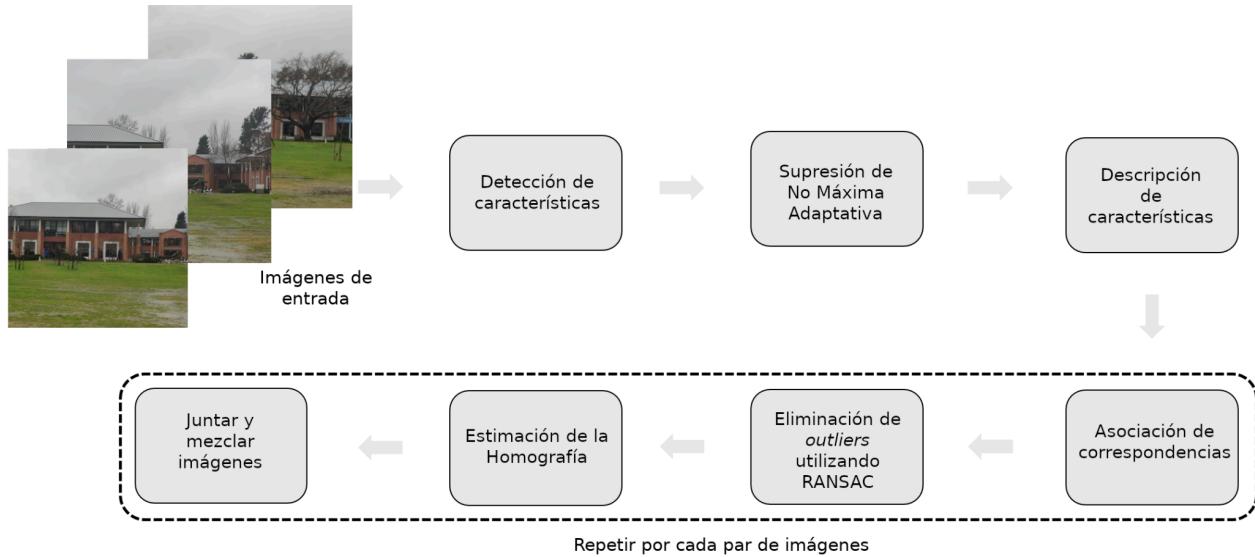


Sugerencia:

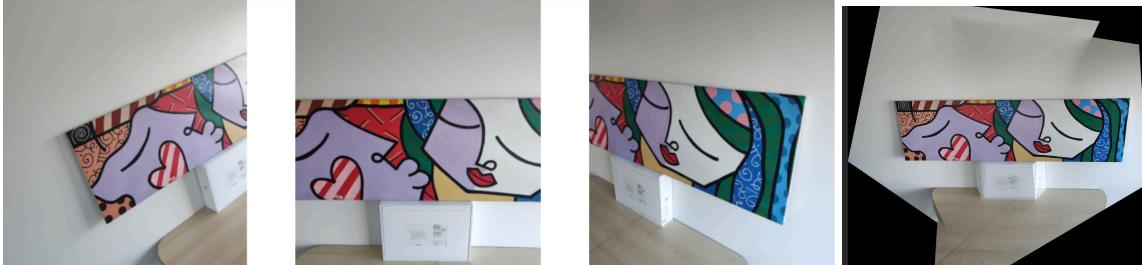
Pueden reducir el tamaño de las imágenes para obtener un conjunto de datos manejable.
Todas las imágenes utilizadas deben estar en jpg.

3. Método de imágenes panorámicas:

A continuación se muestra el esquema general del *pipeline* de procesamiento:



Tres imágenes de ejemplo y un resultado esperado:



3.1. Detección y descripción de características visuales (*features*):

En la mayoría de las tareas de visión por computadora, el primer paso para unir un panorama es detectar y extraer características visuales distintivas.

Pueden seleccionar el tipo de características visuales que prefieran para detectar y extraer de las imágenes. Pueden utilizar las funciones de OpenCV para esto, tal como fue visto en las clases tutoriales.

3.2. Supresión de No Máxima Adaptativa (*Adaptive Non-Maximal Suppression*, ANMS)

En una imagen real, una característica visual nunca es perfectamente nítida, cada una puede recibir muchas “respuestas” por parte de los detectores de características visuales, además, para la tarea que queremos realizar **nos interesa que las características visuales estén “bien” distribuidas en la imagen.**

Este paso tendrá entonces dos objetivos:

1. Definir las coordenadas de máxima respuesta local para el detector de características utilizado. Pueden utilizar los enfoques vistos en las clases tutoriales.
2. Acotar la cantidad máxima de características visuales que se utilizarán de una imagen quedándonos con las N mejores que maximicen, además, la distancia entre ellas.

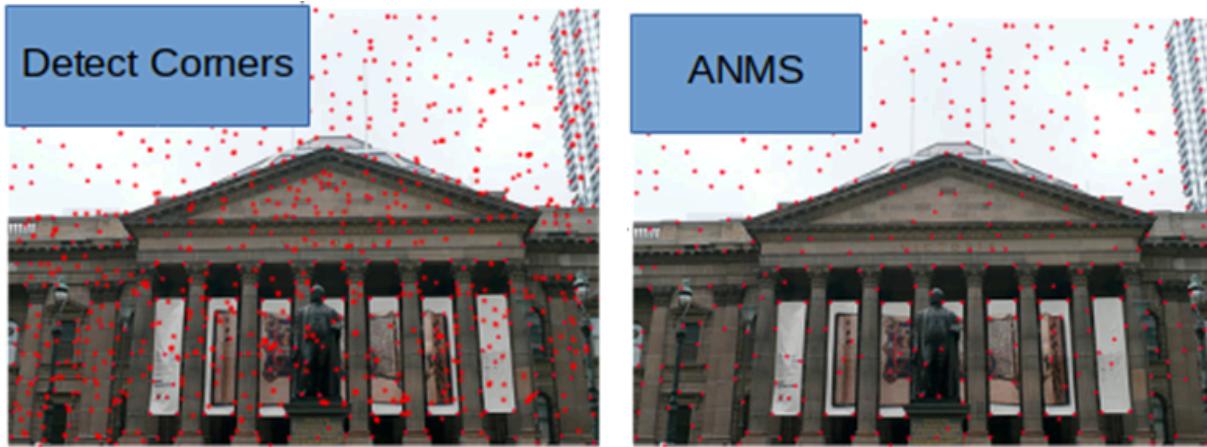
A continuación se muestra un pseudocódigo posible para implementar este comportamiento:

Algorithm 1 Supresión No Máxima Adaptativa, ANMS

Input: Keypoints $K = \{k_1, \dots, k_n\}$ tal que $k_i = (x_i, y_i, r_i)$ donde (x_i, y_i) refiere a su coordenada y r_i a su respectiva respuesta en la imagen I y N máximo número de keypoints requeridos en toda la imagen.

- 1: Definir un conjunto K_l de máximos locales considerando una vecindad de tamaño fijo alrededor de cada keypoint k_i .
 - 2: $\forall k_i \in K_l$ local maximal, inicializar una respuesta global $R_i = \infty$
 - 3: **for** $i = [1 : |K_l|]$ **do**
 - 4: **for** $j = [1 : |K_l|]$ **do**
 - 5: **if** $r_j > r_i$ **then**
 - 6: $SD = (x_j - x_i)^2 + (y_j - y_i)^2$
 - 7: **end if**
 - 8: **if** $SD < R_i$ **then**
 - 9: $R_i = SD$
 - 10: **end if**
 - 11: **end for**
 - 12: **end for**
 - 13: Ordenar R_i de forma descendente y elegir los N mejores.
-

Ejemplo de la aplicación del método ANMS sobre un conjunto de características visuales detectadas. Notar que quedan en la imagen una menor cantidad de características mejor distribuidas:



3.3. Asociación de características (*Matching*):

En este paso queremos encontrar buenas correspondencias entre los descriptores de las diferentes imágenes.

La asociación de características se realiza entre pares de imágenes, para cada descriptor de la imagen 1 buscaremos los dos vecinos más cercanos entre los descriptores de la imagen 2.

Para esto pueden usar el `cv2.BFMatcher` o el `cv2.FlannBasedMatcher`.

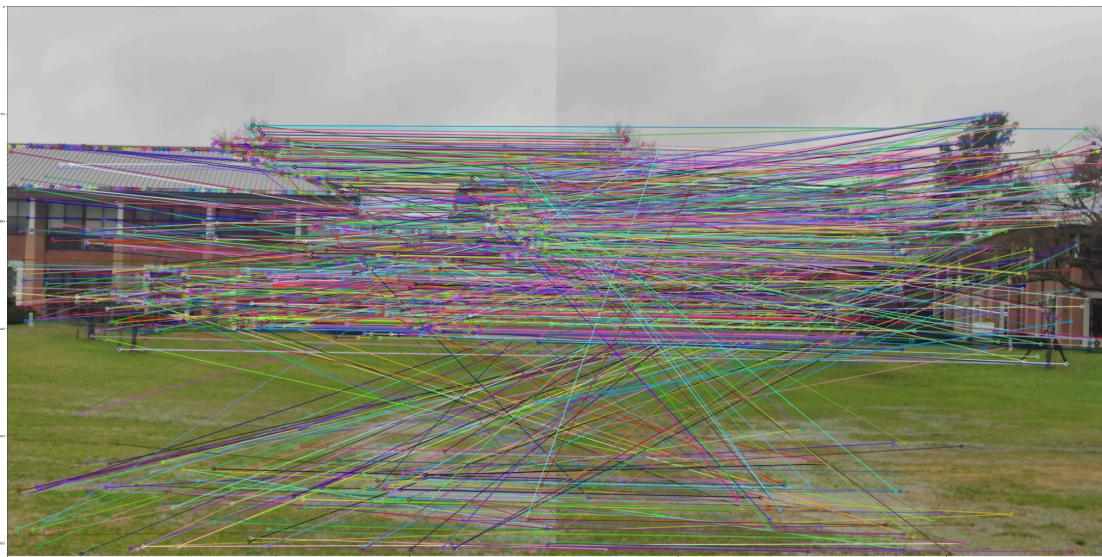
En base a esto deberán realizar la asociación de correspondencias entre las tres imágenes de la panorámica.

Se recomienda definir una imagen como “ancla”, a la cual se le buscará relacionar y “juntar” las dos imágenes restantes.

Deberán realizar la búsqueda de correspondencias aplicando:

- Verificación cruzada (*cross-check*): Una asociación es considerada válida sólo si obtenemos el mismo par de asociaciones en “ambas direcciones”, es decir, cada keypoint es la mejor asociación del otro. **Verificar documentación de OpenCV para esto.**
- Aplicar la comprobación de Lowe (*Lowe's ratio*): Quedarse solamente con aquellas correspondencias cuyo segundo vecino más cercano está al menos a más distancia que un determinado threshold. Recordar que la distancia entre características visuales depende de su representación, características binarias (BRIEF, ORB) utilizan distancia Hamming mientras que SIFT utiliza norma 2.

Resultado posible de realizar la búsqueda de correspondencias:



3.4. Estimación de la homografía “manualmente”

En esta etapa se buscará estimar una homografía entre la imagen central o “ancla” y alguna otra determinando manualmente 4 pares de correspondencias (pueden utilizar un editor de imágenes para determinar las coordenadas de los puntos de interés).

Luego deberán aplicar DLT para hallar la homografía con estos puntos seleccionados. Deberán realizar esto para los dos pares de imágenes.

Deben resolver el algoritmo de DLT (*Direct Linear Transformation*) ustedes sin utilizar la librería OpenCV.

3.5. RANSAC para eliminación de *outliers* y estimación de homografías

Nuevamente se buscará estimar una homografía entre la imagen central o “ancla” y alguna otra, pero esta vez de manera algorítmica utilizando RANSAC.

Algorithm 2 Homografía utilizando RANSAC

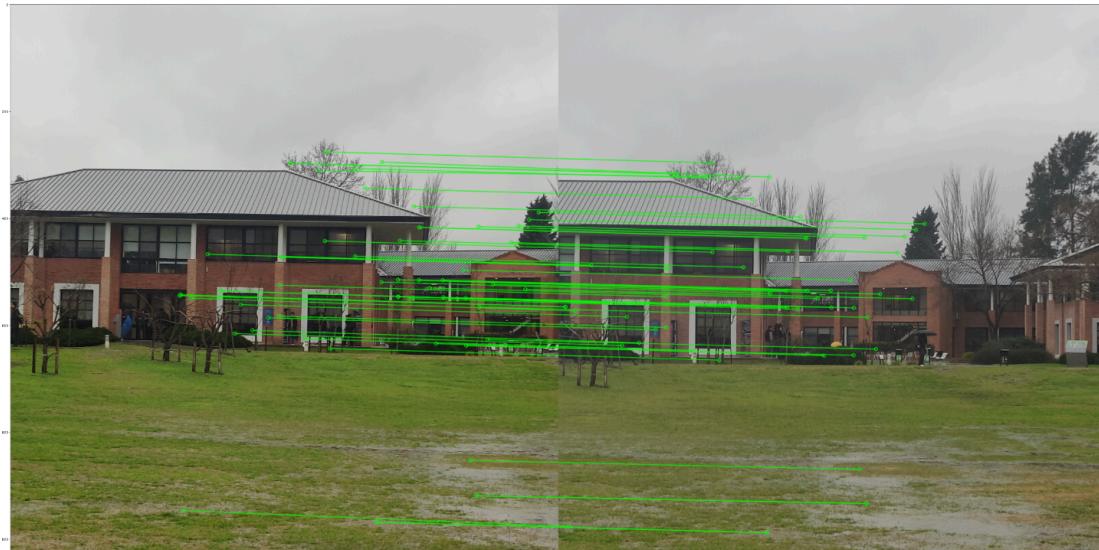
Input: Pares de correspondencias $\{k_i, k_j\} \in C_{I_1, I_2}$ donde k_i es un keypoint detectado en la imagen I_1 y k_j es un keypoint detectado en la imagen I_2 , T cantidad de iteraciones de RANSAC a realizar.

Output: Homografía H y el conjunto de correspondencias *inliers*.

- 1: **for** $i = [1 : T]$ **do**
 - 2: Seleccionar 4 pares de correspondencias k_i, k_j aleatorias de C_{I_1, I_2}
 - 3: Calcular homografía H utilizando los pares de keypoints seleccionados
 - 4: Determinar correspondencias *inliers* $\{k_i, k_j\}$ tal que la distancia $\sqrt{(k_j - H \cdot k_i)^2} < t$, con t un umbral en píxeles.
 - 5: Recordar el conjunto de correspondencias *inliers* más grande
 - 6: **end for**
 - 7: Recalcular la homografía H con cuadrados mínimos utilizando todas las correspondencias *inliers*
-

Pueden utilizar `cv2.findHomography` para el cálculo de la homografía final utilizando el conjunto de correspondencias *inliers*.

Ejemplo de correspondencias *inliers* detectadas utilizando RANSAC:



3.6. Juntando y mezclando imágenes (*Stitching and Blending*)

Para juntar las imágenes relacionadas primero es necesario calcular el tamaño final de la imagen panorama. Es posible lograr esto, inspeccionando las coordenadas máximas y mínimas que se obtienen al transformar las imágenes por la homografía H.

Si vamos a pegar múltiples imágenes hay que tener en cuenta todos los vértices y dónde irán a parar con todas las H computadas. También hay que tener en cuenta las traslaciones necesarias que deben agregarse a las homografías para que el stitching final quede correcto.

Para cada imagen realizamos el *warping* hacia la imagen “ancla”. Usando las homografías encontradas, pueden utilizar las funciones de OpenCV `cv2.perspectiveTransform` y `cv2.warpPerspective` para esto.

Finalmente, pueden aplicar una corrección de color realizando una mezcla (*blending*) de los canales de color de las imágenes unidas. Pueden crear una máscara de “degradé” para cada imagen utilizando `cv2.distanceTransform` la cual llamaremos W. Es posible mezclar un canal de color aplicando la correspondiente máscara W a cada imagen y luego uniendolas todas utilizando la siguiente ecuación:

$$I = \frac{\sum_i W_i \cdot I_i}{\sum_i W_i}$$

Ejemplo de máscaras aplicadas a cada imagen a un canal de color:



Para el resultado final en color, deberemos realizar este procedimiento para cada canal red, green y blue de manera independiente.

Resultados finales de los conjuntos de imágenes de ejemplo:



4. Informe y grupos de trabajo

Deberán presentar un informe de **no más de 6 páginas** exponiendo las técnicas y decisiones tomadas a lo largo del trabajo. El trabajo es grupal, en **grupos de dos personas**.

El informe debe tener al menos tres secciones, **introducción, método y resultados**. Deberán explicar el método implementado detallando detectores, descriptores y técnicas utilizadas. El nivel de detalle debe ser acorde con la extensión máxima del informe considerando que deben exponer imágenes de los resultados obtenidos en cada etapa del *pipeline* de procesamiento.

Deberán entregar el informe junto con el código utilizado para resolver el problema y los conjuntos de imágenes propuestos por ustedes. **Todas las imágenes deben ser .jpg con una calidad máxima de 85** (en general, 1,5Mb por imagen).

Pueden trabajar sobre cualquiera de los conjuntos de imágenes de prueba provistos (o con ambos), y deberán exponer resultados finales en un conjunto de imágenes propuesto por ustedes.

Se sugiere realizar una vista panorámica de alguna porción de la universidad, ¡para ayudar a nuestro fotógrafo frustrado!