

Tarea 03

David Rojas

October 2019

1 Resumen

Se utilizó una implementación de SIFT en conjunción con una implementación artesanal de RANSAC para realizar stitching de imágenes.

El propósito de esto es crear una aplicación para generar imágenes panorámicas.

2 Introducción

Este proyecto busca generar imágenes estilo panorámicas usando Stitching con descriptores locales comunes entre dos imágenes para generar una única imagen mas amplia. Para hacer este Stitching necesitamos buscar una transformación entre las dos imágenes tal que lleve los descriptores de una a los descriptores equivalentes en la otra. Para este proceso utilizamos RANSAC. Los descriptores locales son producidos utilizando SIFT.

A continuación se explica el funcionamiento de Stitching y RANSAC.

1. **Stitching:** El stitching es un proceso mediante el evaluamos los matching descriptors y calculamos una homografía que lleve de un set de descriptores al otro. Luego se utiliza esta homografía para llevar la segunda imagen al mismo espacio de la primera.
2. **RANSAC:** Para realizar el RANSAC tomamos todos los puntos que fueron evaluados como matches y buscamos una transformación homográfica que nos lleve de un set a la mayor cantidad de puntos del otro set como sea posible. Para esto, busca una serie de puntos, busca la transformación asociada y calcula cuantos puntos son correctamente transformados utilizando la. El ciclo de búsqueda acaba cuando se encuentra una transformación satisfactoria o se alcance un determinado numero de iteraciones. El propósito de esto es intentar generar un modelo que se ajuste a los datos sin vernos muy afectados por los datos outliers.

3 Desarrollo

Para SIFT se utilizo la implementación de OpenCV. Luego se buscan los matches utilizando la implementación de OpenCV nuevamente y se filtran los malos

matches.

A continuación, generamos dos listas de puntos sin elementos repetidos (Nuestro algoritmo de transformaciones necesita puntos de la forma $(x, y, 1)$, por lo tanto los agregamos a la lista en ese mismo formato).

Luego usamos estas dos listas de puntos (tales que los puntos de una imagen probablemente se correspondan al punto en la misma posición en la otra lista) para correr el algoritmo de RANSAC. Limitamos el numero de iteraciones de RANSAC a 10000, el threshold a 4 y el ratio a 0.6.

RANSAC luego elige 3 elementos de las listas (sin repetición) e intenta generar una transformación entre ellos. Luego calcula esta transformación para todos los elementos de la segunda lista, y compara cuantos puntos en la segunda lista se encuentran a lo mas a 4 unidades de distancia del punto que se supone de origen. Si encontramos que el 60 por ciento de los puntos encuentran la transformación aceptable, retornamos la transformación encontrada. De caso contrario, seguimos iterando hasta encontrar una transformación aceptable o hasta alcanzar las 10000 iteraciones.

Una vez obtenemos la transformación, procedemos a buscar los limites de nuestra nueva imagen (Buscamos los x limites, es decir, mas bajo y mas alto. Hacemos lo mismo para y) y usamos esta información para generar una imagen completamente nueva. Vamos generando esta imagen píxel a píxel, consultando cada vez si el píxel tiene alguna correspondencia en la imagen original. Luego consultamos la posición de este nuevo píxel con respecto al espacio de la segunda imagen. Si encontramos una equivalencia aceptable en la segunda imagen, hacemos una interpolación bilineal para calcular el promedio de colores de los píxeles que rodean al píxel correspondiente a nuestro nuevo píxel. Píxeles que no se corresponden a ningún píxel en la imagen original o la segunda imagen se dejan en negro.

Una vez la imagen esta completada, se escribe con el nombre de "panoramica.png".

4 Resultados Experimentales y Discusión

El proceso toma bastante tiempo, con resultados bastante prometedores en algunos casos pero poco prometedores en otros.

El caso 4 fue el que obtuvo mejores resultados, con un fit casi perfecto y condiciones ideales, con las fotos tomadas una directamente junto a la otra.

Los casos 1 y 2 obtuvieron resultados similares entre si, en que el fit es excelente, pero debido a diferencias en la disposición de los elementos en las imágenes, estos aparecen duplicados en las imágenes.

Los casos 3 y 5 tienen fits aceptables, pero el caso 5 los cambios de ángulo generan diferencias completamente irreconciliables, lo que genera un efecto poco agradable.

El caso 8 presenta un cambio de ángulo lo suficientemente grande para causar un mal fit, pero en general encuentra algunas cosas que reconciliar.

El caso 7 es un cambio de ángulo tan grande que el algoritmo simplemente no puede encontrar una equivalencia.

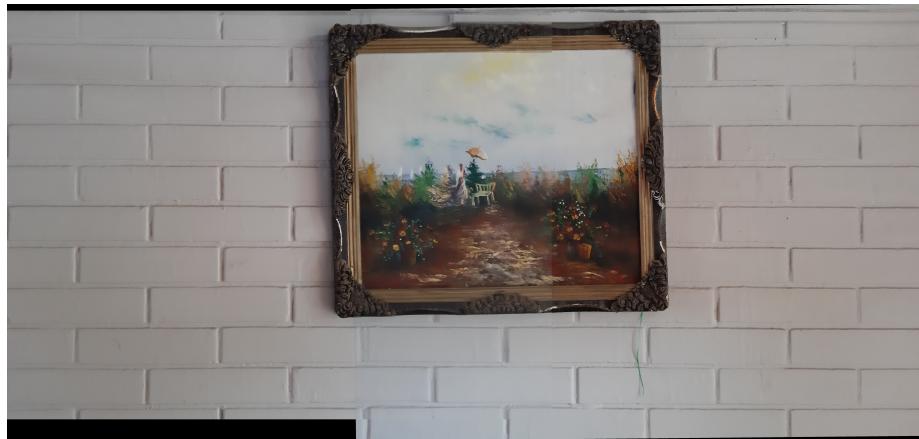


Figure 1: Caso 4.

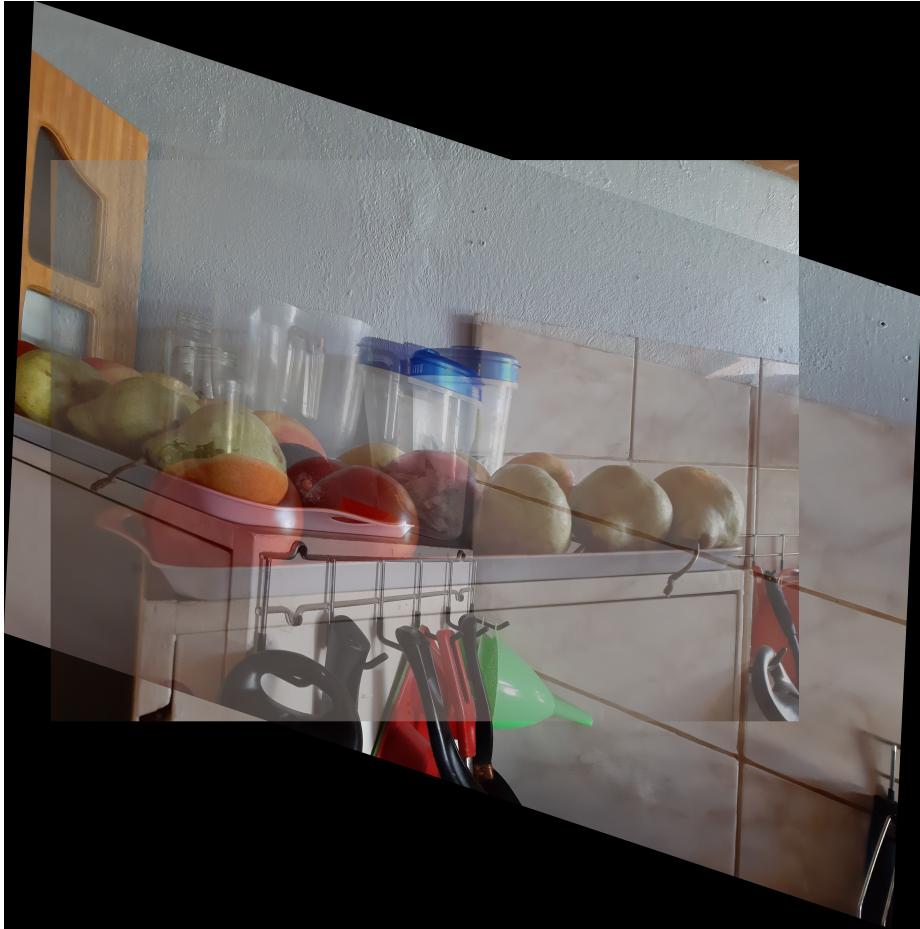


Figure 2: Caso 7.

5 Conclusión

Este algoritmo para generar imágenes panorámicas funciona bien para imágenes en que los objetos no cambian de posición con respecto a otros objetos en la escena, es decir, imágenes tomadas muy de cerca con objetos a diferente distancia de la cámara no suelen funcionar correctamente. Sin embargo, se comporta muy bien respecto a rotaciones que mantengan ángulos ortogonales con respecto la normal de la primera imagen, pero claramente no puede reconciliar imágenes donde los elementos cambian de posición relativa entre si.

El caso 3 es bastante problemático de analizar, pero lo mas probable es que haya demasiados descriptores los arboles, por lo que el algoritmo intenta cuadrarlos con mayor importancia que el resto de la imagen.