Trabajo Practico 3
Segunda entrega
Robótica Móvil

Franco Ignacio Vallejos Vigier
Facundo Emmanuel Messulam

14 de febrero de 2024

4. Calibración)

Adoptamos como dato de calibración los proporcionados en la rosbag, entonces de estas dos cámaras obtenemos las matrices intrínsecas que describen las propiedades individuales de la cámara como lo es el centro óptico por ejemplo, como segundo resultado de la calibración capturamos los factores de distorsión de las cámaras que describen las distorsiones que produce el lente de la cámara a la imagen resultante (ya sea radial o tangencial), como ultimo retorno de la calibración obtenemos la matriz extrínseca (roto-traslacional).

Podemos encontrar estas matrices y factores en los `/calibration_data/left.yaml` y `/calibration_data/right.yaml`

5, 6. Rectificación y Sincronización)

Para la rectificación en ROS 2 podemos usar fácilmente el paquete `stereo_image_proc` con este paquete vamos a tener nuevos tópicos donde se van a publicar las imágenes rectificadas y la información de las cámaras. Estos tópicos son `/left/image_rect` y `/right/image_rect` . Para que el proyecto use estas imágenes creamos los subscriptores a estos tópicos, por ultimo correspondemos las imágenes de la cámara izq. y der. con el objeto TimeSynchronizer. El resultado de la rectificación de las imágenes la guardamos en `/catkin_ws/src/result/` , las imágenes con el mismo índice son las imágenes que están rectificadas.

7. Extraer Features Visuales: Keypoints y descriptores)

Para la extracción de keypoint y descriptores usamos directamente como detector de keypoint y descriptores SIFT (Scale-Invariant Feature Transform). Por lo tanto lo que hacemos es operar con diferencias gaussianas para detectar puntos con cambios grandes de intensidad, los puntos extremos (cercanos a los mínimos y máximos son grandes candidatos para ser keypoints), luego se filtran los puntos que tienen suficiente contraste.
Los keypoint de las imágenes podemos encontrarlas en `/catkin_ws/src/result/keypoints`. Mientras que las orientaciones de los keypoints y la orientación dominante esta en la carpeta `/catkin_ws/src/result/SIFT Frames`.

8. Matches y correspondencias visuales)

Para obtener los matches SIFT de dos imágenes tenemos que comparar los descriptores de los keypoints utilizando medidas de similitud como puede ser la distancia Euclidiana (y si los descriptores son binarios se puede usar otras medidas de distancia como Hamming). Estos descriptores son una representación numérica de la región circundante al keypoint, se usa para çuantificar"las características visuales de la región y permiten la comparación entre keypoints. En el trabajo practico además hacemos un filtrado de los descriptores de la imagen izq. y der. para obtener los "good matchesçomo aquellos que cumplen el ratio factor en este caso 0.99 podemos ajustarlo aun mas para filtrar las correspondencias basadas en distancias relativas. Esto ultimo se debe al Ratio Test, por ende teniendo dos descriptores si son correspondientes la relación que tiene que haber entre sus distancias tiene que ser mayor que si no lo son.
Los matches de las imágenes están en `/catkin_ws/src/result/matches/allMatches`.
El filtrado por distancia lo encontramos en `/catkin_ws/src/result/matches/matchesFilteredByDistance`.

9. Triangulación)

En la triangulación en las cámaras stereo tenemos que determinar la posición tridimensional de un punto teniendo las proyecciones en el frame de las 2 cámaras stereo. Dado que tenemos ya las correspondencias entre los frames izq. y der. además de las matrices intrínsecas y extrínsecas para describir la relación con la geometría epipolar entre las dos cámaras con todo esto ahora podemos hacer triangulación por mínimos cuadrados así podemos obtener las coordenadas homogéneas para obtener las coordenadas 3D. Después de esto normalizamos las coordenadas.

10. Filtrar de Correspondencias Espúreas)

Con los frames de ambas cámaras calculamos la matriz homográfica, esta matriz homográfica nos permite hacer una transformación entre la correspondencia de los keypoint de un frame de la cámara izq al frame de la cámara der. esto es así porque dicha matriz describe la proyección de la perspectiva entre dos planos de imagen, se utiliza de modo de alinear las dos imágenes rectificadas de manera de que correspondan punto a punto. A esta matriz H la calculamos usando RANSAC (Random Sample Consensus), este algoritmo lo que va a hacer es corresponder los keypoints de las dos imágenes rectificadas incluso se hay outliers o datos raros". Luego obtenida la matriz H podemos usarla para transpolar los puntos de un frame a otro luego (en este caso de frame izq a der si queremos el camino inverso tenemos que invertir la matriz 3x3 que es la H).
El resultado de la transformación de los keypoint izq. al frame der. usando la H lo podemos ver en `/catkin_ws/src/result/keypoints transform L->R/`

11. Computar el Mapa de Disparidad)

En este punto necesitamos calcular el mapa de disparidad que es una representación de la diferencia de profundidad entre dos frames de nuestras 2 cámaras stereos. Como nuestras 2 cámaras capturan la misma escena pero con una ligera diferencia en la baseline entonces la disparidad entre esas 2 imágenes de la misma escena representa el desplazamiento horizontal de un punto en una imagen con respecto al mismo punto en la otra. Además sabemos que la relación entre la disparidad y la distancia es inversamente proporcional, entonces, los objetos mas cercanos tendrán disparidad mayor. Los disparity maps se pueden encontrar en `/catkin_ws/src/result/disparityMaps/`

12. Reconstruir la Escena 3D de manera Densa)

Hacemos la reconstrucción 3D de la escena capturada por nuestro par de cámaras estereo usando el mapa de disparidad anterior y la matriz Q para obtener la transformación entre disparidad y distancia (inversamente proporcional) esta matriz Q en su ultima columna va a tener un vector columna con : las coordenadas del punto principal de la cámara stereo izq, la distancia focal f y la resta entre la x de la cámara izq y la x de la cámara derecha sobre la baseline. De igual forma como toda dense cloud la publicamos por el tópico así la podemos visualizar con RViz.

13. Estimar la Pose utilizando Visión Monocular)

Para la estimación de pose monocular lo que hacemos es recuperar la pose usando la matriz esencial y los keypoint del frame de la cámara izq. para obtener R y T de la nueva posición. Básicamente acumulamos tanto la traslación como la orientación así conformamos la nueva pose y ploteamos en 2D el acumulado (se puede ver el resultado en `/catkin_ws/src/result/Estimated Trajectory LeftCam/`) donde se gráfica el trayectoria realizada por la camara hasta el momento del frame analizado. Mirando las imágenes decidimos plotear el resultado en 2D porque la cámara no realiza un gran movimiento en el eje Z (la cámara rota, haciendo que el movimiento que queremos estimar se encuentre principalmente en el eje X). Publicamos la pose de la cámara izq. por el tópico `/pose_camera_left`