

Práctica 1. Cálculo del egomotion de una cámara RGBD

Miguel Cazorla, Vicente Morell, Diego Viejo
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

27 de febrero de 2014

En esta segunda práctica, se pretende realizar la construcción de un mapa del entorno a partir de los datos capturados por una cámara RGBD (en nuestro caso, usaremos una Kinect). Para ello usaremos ROS y la librería de procesamiento 3D Point Cloud Library (PCL).

Imaginemos que tenemos un robot que se está moviendo por el entorno. Cada cierto tiempo, el robot realiza una toma de datos de la cámara. Esta cámara nos proporciona una imagen RGB y una nube de puntos 3D. En la Figura 1 se puede ver un par de ejemplos de imágenes 3D. El objetivo de esta práctica es, usando los datos en cada posición del robot, calcular el movimiento (transformación 3D) que ha realizado el robot (le llamaremos *egomotion*) para cada par de imágenes consecutivas. Una vez calculado este movimiento, podemos transformar todos los datos capturados en un sistema de coordenadas común y así podemos construir un mapa. Como los datos 3D capturados son numerosos, habrá que usar alguna técnica para reducirlos.



Figura 1: Dos ejemplos de datos 3D.

1. Proceso a seguir

Vamos a detallar qué pasos tenemos que seguir para realizar la práctica. Vamos a suponer que el robot ha realizado t tomas de datos, por lo que tendremos

un conjunto de datos $X = x_1, x_2, \dots, x_t$. Cada dato x_i es un conjunto de puntos 3D. Los pasos a seguir son los siguientes:

Entrada: Conjunto de datos X .

- 1: Transformación total $T_T = I$
- 2: Mapa 3D M
- 3: **para** $i=1$ hasta t **hacer**
- 4: Extraer características $C_i = \{c_1^i, c_2^i, \dots, c_m^i\}$ de los datos x_i .
- 5: Extraer características $C_{i+1} = \{c_1^{i+1}, c_2^{i+1}, \dots, c_k^{i+1}\}$ de los datos x_{i+1} .
- 6: Encontrar emparejamientos $P = \{p_1, p_2, \dots, p_l\}$ entre las características C_i y C_{i+1} . Cada emparejamiento es un par de características de cada conjunto de datos $p_j = \{c_a^i, c_b^{i+1}\}$
- 7: Obtener la mejor transformación T_i que explican los emparejamientos.
- 8: Obtener la transformación total $T_T = T_T * T_i$.
- 9: Aplicar al conjunto de datos C_{i+1} la transformación T_T para colocarlos en el sistema de coordenadas de C_1 . Acumular el conjunto de datos transformados en M .
- 10: **fin para**
- 11: **devolver** M

Vamos a ir explicando en detalle los pasos a seguir.

1. Extracción de características. Este paso nos devolverá un conjunto de características C_i que será el resultado de aplicar un detector y un descriptor de características. En clase de prácticas explicaremos los disponibles. Habrá que hacer un estudio (ver el apartado de Experimentación) para determinar cuál es el más adecuado (por tiempo de ejecución y eficacia).
2. Encontrar emparejamientos. Usaremos el método que proporciona PCL para encontrar las correspondencias. El resultado de este paso es un conjunto de emparejamiento.
3. Como hemos comentado antes, es posible que haya muchos malos emparejamientos. En este paso tenemos que determinar la mejor transformación que explica los emparejamientos encontrados. Para ello, usaremos el algoritmo RANSAC.
4. Por último, hay que construir el mapa. Como cada toma de la Kinect tiene aproximadamente 300.000 puntos, en el momento que tengamos unas cuantas tomas vamos a manejar demasiados puntos, por lo que hay que proceder a reducirlos. Para ellos, podemos usar el filtro de reducción VoxelGrid, disponible en PCL.

2. Experimentación

Hay que realizar un estudio de los distintos detectores y descriptores disponibles en PCL. En clase de prácticas os iremos dando detalles de cuáles podéis usar. Habrá que calcular el tiempo de cálculo de dichos detectores y descriptores para determinar cuál es más eficiente. También tenéis que determinar cuál de ellos es más eficaz, es decir, cuál consigue mejores resultados. También se usarán el mejor método que hayáis obtenido en la anterior práctica para compararlo con los 3D.

3. Parte optativa

Como parte optativa de esta práctica se plantea aplicar el método desarrollado con una cámara Kinect en tiempo real. O bien hacerlo con la Kinect encima de un robot.

4. Documentación a entregar

La documentación de la práctica es una parte muy importante en la puntuación final (un 60 % de ella). El código debe estar debidamente comentado, indicando qué se hace en cada punto. Además, se debe entregar una documentación (cualquier formato: PDF, HTML, etc.) con los siguientes puntos:

1. Descripción de lo que se pretende hacer. Resumen del conjunto de la práctica.
2. Descripción de cómo se ha realizado la práctica (consideraciones, problemas encontrados, etc.)
3. Ejemplo de ejecución desde varias posiciones. Al menos documentar y mostrar tres ejecuciones desde distintos puntos del entorno, para demostrar el buen funcionamiento de vuestra implementación.

Normas de entrega de la práctica:

- La práctica se entregará antes de las 24 horas del viernes 23 de Mayo del 2014.
- La entrega se realizará a través del Moodle de la asignatura.