

## Trabajo Práctico N° 2 (Parte 1): **Extracción de líneas mediante RANSAC**

### Ejercicio 1:

#### **¿Do do dónde está la recta?**

- I. Escribir el código para una función que implemente el método de hipótesis/test iterativo RANSAC [1, cap. 6.1]. El formato de la función debe ser

`y=ransac(x,N,n,d,t)`

donde

`x`: matriz de puntos de entrada.

`N`: número de iteraciones del algoritmo.

`n`: cantidad mínima de puntos para instanciar el modelo.

`d`: umbral de aceptación de inliers (punto ajustado correctamente).

`t`: #conjunto de consenso para la de aceptación del modelo ajustado.

`y`: contiene los parámetros del mejor modelo ajustado y vector de índice de los inliers.

- II. Usar la función `ransac()` para procesar el conjunto de puntos generados por TP2\_ejercicio1.p<sup>1</sup>. Evaluar el comportamiento del algoritmo respecto a `N`, `d`, y `t`.

### Ejercicio 2:

#### **Con una no me alcanza...**

- I. Analizar los puntos generados por TP2\_ejercicio2.p<sup>2</sup>. y emplear `ransac()` para escribir el código de una nueva función `multi_ransac()`, que permita ajustar los modelos presentes.

### Ejercicio 3:

#### **Volviendo al primer amor...**

- I. Aplicar el algoritmo modificado `multi_ransac()` para ajustar las líneas presentes en los diferentes escaneos del LIDAR del TP1.
- II. Explorar otras alternativas como [2] para el ajuste de rectas en presencia de outliers. Elegir una y escribir su código. Comparar su desempeño respecto a `multi_ransac()`.

**Comandos de Matlab:** *help, doc, randperm, tic, toc.*

---

1Ejecutar TP2\_ejercicio1 para obtener información de las variables de entrada y salida de la función.

2Ejecutar TP2\_ejercicio2 para obtener información de las variables de entrada y salida de la función.

Referencias:

- [1] Szeliski R. 2010. **Computer Vision: Algorithms and Applications**. Springer-Verlag: London.
- [2] Nguyen, V., Martinelli, A., Tomatis, N., & Siegwart, R. (2005, August). **A comparison of line extraction algorithms using 2D laser range finder for indoor mobile robotics**. In *Intelligent Robots and Systems, 2005 (IROS 2005)*. 2005 IEEE/RSJ International Conference on (pp. 1929-1934).

Trabajo Práctico N° 2 (Parte 2):

***Extracción puntos característicos en imágenes para estimar transformaciones***

Ejercicio 4:

- I. Utilizar la función TP2\_ejercicio4.p para generar una imagen rotada y escalada a partir de una imagen de su elección. La matriz RGB de la imagen a transformar es el único argumento de la función mencionada.
- II. Implemente una función que utilice RANSAC para hallar la matriz de transformación entre la imagen original y la imagen generada en el punto anterior. Para esto debe utilizar los puntos de interés extraídos de cada imagen, sus descriptores, y sus correspondencias. Se utilizarán características del tipo esquina (con el detector de Harris) o del tipo SURF mediante la correspondiente selección. Para mayores referencias tomar como base el ejemplo de MatLab *"Find Image Rotation and Scale Using Automated Feature Matching"*. En otras palabras, lo que se pretende es que reemplace el punto 4 (step 4) del ejemplo por una función hecha por usted mismo.
- III. Repita el punto I varias veces y compare las imágenes originales con las imágenes transformadas mediante la matriz hallada. Defina una métrica para realizar esta comparación. También compare la performance de su algoritmo frente a la función de MatLab que usa el ejemplo.

**Algunos comandos de Matlab:** *detectSURFFeatures*, *visión.CornerDetector*, *extractFeatures*, *matchFeatures*