

비주얼오도메트리와증강현실  
RANSAC simulation for line fitting  
( AIE 6660-01 )

Due: 2023년, 10/16일

학번 : 120230230

이름 : 김정현

```

import numpy as np
import matplotlib.pyplot as plt

def fit_line(x1, y1, x2, y2):
    """Fit a line using two points."""
    a = (y2 - y1) / (x2 - x1)
    b = y1 - a * x1
    return a, b

def compute_error(a, b, x, y):
    """Compute the error of the line model."""
    y_pred = a * x + b
    return abs(y_pred - y)

def introduce_outliers(x, y, outlier_ratio):
    """Introduce outliers into the dataset based on the given outlier ratio."""
    n_outliers = int(n_points * outlier_ratio / 100)
    indices = np.random.choice(n_points, n_outliers, replace=False)
    y_outliers = np.random.uniform(0, 2000, n_outliers)
    y[indices] = y_outliers
    return x, y

def ransac(x, y, max_iterations=1000, threshold=10):
    best_a = None
    best_b = None
    best_inliers = 0
    best_inlier_mask = None

    for _ in range(max_iterations):
        idx = np.random.choice(len(x), 2, replace=False)
        x_sample, y_sample = x[idx], y[idx]
        a, b = fit_line(x_sample[0], y_sample[0], x_sample[1], y_sample[1])
        errors = compute_error(a, b, x, y)
        inliers = np.sum(errors < threshold)

        if inliers > best_inliers:
            best_a, best_b = a, b
            best_inliers = inliers
            best_inlier_mask = errors < threshold

    return best_a, best_b, best_inlier_mask

a_original = 2
b_original = 200
n_points = 300
x = np.linspace(0, 4000, n_points)
noise = np.random.normal(0, 1, n_points)
y = a_original * x + b_original + noise

outlier_ratios = [10, 30, 50, 70]
datasets = [introduce_outliers(x.copy(), y.copy(), ratio) for ratio in outlier_ratios]

results_with_outliers = [ransac(x_data, y_data, max_iterations=300, threshold=100) for x_data, y_data in datasets]

plt.figure(figsize=(15, 10))

<Figure size 1500x1000 with 0 Axes>
<Figure size 1500x1000 with 0 Axes>

for i, (x_data, y_data) in enumerate(datasets):
    a, b, inlier_mask = results_with_outliers[i]
    outlier_mask = np.logical_not(inlier_mask)
    line_x = np.arange(0, 4000)
    line_y_custom_ransac = a * line_x + b

    plt.subplot(2, 2, i+1)
    plt.scatter(x_data[inlier_mask], y_data[inlier_mask], s=10, color='blue', label='Inliers')
    plt.scatter(x_data[outlier_mask], y_data[outlier_mask], s=10, color='red', label='Outliers')
    plt.plot(line_x, line_y_custom_ransac, color='green', label='Custom RANSAC line (max_iterations=300)')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.title(f'Custom RANSAC Result for {outlier_ratios[i]}% Outliers')
    plt.xlim(0, 4000)
    plt.ylim(0, 2000)
    plt.legend()
plt.tight_layout()
plt.show()

```

