

EXPERIMENT 3

2D Transformations – Translation, Scaling, and Rotation

Code:

```
import numpy as np import matplotlib.pyplot as plt
```

----- Transformation Functions -----

```
def translate(points, tx, ty): """Translate by (tx, ty)""" T = np.array([[1, 0, tx], [0, 1, ty], [0, 0, 1]]) return (T @ points.T).T
```

```
def scale(points, sx, sy): """Scale by (sx, sy)""" S = np.array([[sx, 0, 0], [0, sy, 0], [0, 0, 1]]) return (S @ points.T).T
```

```
def rotate(points, angle_deg): """Rotate by angle (in degrees) counterclockwise""" theta = np.radians(angle_deg) R = np.array([[np.cos(theta), -np.sin(theta), 0], [np.sin(theta), np.cos(theta), 0], [0, 0, 1]]) return (R @ points.T).T
```

----- Visualization -----

```
if name == "main": # Define a triangle polygon polygon = np.array([[2, 1, 1], [5, 1, 1], [3, 4, 1], [2, 1, 1]]) # closed shape
```

```
# Apply transformations
translated = translate(polygon, 4, 3)
scaled = scale(polygon, 2, 1.5)
rotated = rotate(polygon, 45)
```

```
# Plot
```

```
plt.figure(figsize=(12, 4))

# Original
plt.subplot(1, 3, 1)
plt.plot(polygon[:,0], polygon[:,1], 'b-',
label="Original")
plt.plot(translated[:,0], translated[:,1], 'r--',
label="Translated")
plt.legend()
plt.title("Translation")
plt.axis("equal")

# Scaling
plt.subplot(1, 3, 2)
plt.plot(polygon[:,0], polygon[:,1], 'b-',
label="Original")
plt.plot(scaled[:,0], scaled[:,1], 'g--',
label="Scaled")
plt.legend()
plt.title("Scaling")
plt.axis("equal")

# Rotation
plt.subplot(1, 3, 3)
plt.plot(polygon[:,0], polygon[:,1], 'b-',
label="Original")
plt.plot(rotated[:,0], rotated[:,1], 'm--',
label="Rotated")
plt.legend()
plt.title("Rotation")
plt.axis("equal")
```

```
plt.tight_layout()  
plt.show()
```