Geometric Transformation of 2D Object

April 2, 2025

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
def create_triangle():
    return np.array([[0, 0], [1, 0], [0.5, 1], [0, 0]])
```

[3]:
```
def translation(shape, tx, ty):
    translation_matrix = np.array([tx, ty])
    return shape + translation_matrix
```

[4]:
```
def rotation(shape, angle_deg):
    angle_rad = np.radians(angle_deg)
    rotation_matrix = np.array([[np.cos(angle_rad), -np.sin(angle_rad)],
                                [np.sin(angle_rad), np.cos(angle_rad)]])
    return np.dot(shape, rotation_matrix.T)
```

[5]:
```
def scaling(shape, sx, sy):
    scaling_matrix = np.array([[sx, 0], [0, sy]])
    return np.dot(shape, scaling_matrix.T)
```

[7]:
```
def plot_shapes(original, translated, rotated, scaled):
    plt.figure(figsize=(6, 6))

    plt.plot(original[:, 0], original[:, 1], 'b-', label='Original')
    plt.plot(translated[:, 0], translated[:, 1], 'r-', label='Translated')
    plt.plot(rotated[:, 0], rotated[:, 1], 'g-', label='Rotated')
    plt.plot(scaled[:, 0], scaled[:, 1], 'm-', label='Scaled')

    plt.axhline(0, color='black',linewidth=0.5)
    plt.axvline(0, color='black',linewidth=0.5)
    plt.grid(color = 'gray', linestyle = '--', linewidth = 0.5)

    plt.legend()
    plt.title('Geometric Transformations')
    plt.axis('equal')
    plt.show()
```

[8]:
```
def main():
    triangle = create_triangle()

    translated_triangle = translation(triangle, 1, 1)
    rotated_triangle = rotation(triangle, 45)
    scaled_triangle = scaling(triangle, 2, 2)

    plot_shapes(triangle, translated_triangle, rotated_triangle,␣
 ↪scaled_triangle)
```

[9]:
```
if __name__ == "__main__":
    main()
```