7 - 8.9. Affine transformations

October 7, 2024

Jarrian Vince G. Gojar
Instructor I
College of Information and Communications Technology, Sorsogon State University, Philippines

1 Introduction

In image processing, `Affine Transformation` refers to a linear mapping method that preserves points, straight lines, and planes. Sets of parallel lines remain parallel after an `affine transformation`. The `affine transformation` technique is typically used to correct for geometric distortions or deformations that occur with non-ideal camera angles.

2 Setup

[ ]:
```
%pip install opencv-python opencv-contrib-python matplotlib
```

3 Initial Setup

[5]:
```
# Import Libraries
import os
import cv2
import numpy as np
import scipy.ndimage as nd
import matplotlib.pyplot as plt

# Asset Root
asset_root = os.path.join(os.getcwd(), '../../assets')

# Image Path
image_path = os.path.join(asset_root, 'images', 'perspective_distortion.jpg')

# Read Image and convert to RGB
input_image = cv2.cvtColor(cv2.imread(image_path), cv2.COLOR_BGR2RGB)

# Display Both Image
plt.figure("Affine Transformation", figsize=(10, 10))
```

```
plt.imshow(input_image)
plt.title("Original Image")
plt.axis('off')

plt.show()
```

4 Affine Transformation in Images

To apply an `affine transformation` to an image, we need to define a `transformation matrix`. The `transformation matrix` is a matrix that describes the transformation to be performed on the input image. The `transformation matrix` is then applied to the input image using the `scipy.ndimage.affine_transform` function. The `scipy.ndimage.affine_transform` function

takes the input image and the `transformation matrix` as input and returns the transformed image.

[7]:
```python
# Affine Transformation Matrix
def affine_transformation_sample(image, radian):
    theta = radian * np.pi / 180
    transformation_matrix = np.array([
        [np.cos(theta), -np.sin(theta), 0],
        [np.sin(theta), np.cos(theta), 0],
        [0, 0, 1]
    ])
    affine_image = nd.affine_transform(image, transformation_matrix)
    return affine_image

# Apply Affine Transformation
affine_image = affine_transformation_sample(input_image, 11)

# Display Both Image
plt.figure("Affine Transformation", figsize=(20, 10))

plt.subplot(1, 2, 1)
plt.imshow(input_image)
plt.title("Original Image")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(affine_image)
plt.title("Affine Transformation")
plt.axis('off')

plt.show()
```

The above code snippet applies an `affine transformation` to the input image by rotating it by 11 degrees. The `affine_transformation_sample` function defines the `transformation matrix` for the rotation and applies the `transformation matrix` to the input image using the `scipy.ndimage.affine_transform` function. The resulting image is then displayed alongside the original image.

5 Summary

- `Affine transformations` are linear transformations that preserve points, straight lines, and planes. They include operations such as rotation, scaling, translation, and shearing.
- `Affine transformations` are used in image processing to correct for geometric distortions or deformations that occur with non-ideal camera angles.
- The `affine_transform` function from the `scipy.ndimage` module can be used to apply `affine transformations` to images.
- The `affine_transform` function requires an input image and a transformation matrix as input arguments. The transformation matrix is a 3x3 matrix that defines the transformation to be applied to the image.

6 References

- Thomas G. (2022). Graphic Designing: A Step-by-Step Guide (Advanced). Larsen & Keller. ISBN: 978-1-64172-536-1
- Singh M. (2022). Computer Graphics and Multimedia. Random Publications LLP. ISBN: 978-93-93884-95-4
- Singh M. (2022). Computer Graphics Science. Random Publications LLP. ISBN: 978-93-93884-03-9
- Singh M. (2022). Computer Graphics Software. Random Publications LLP. ISBN: 9789393884114
- Tyagi, V. (2021). Understanding Digital Image Processing. CRC Press.
- Ikeuchi, K. (Ed.). (2021). Computer Vision: A Reference Guide (2nd ed.). Springer.
- Bhuyan, M. K. (2020). Computer Vision and Image Processing. CRC Press.
- Howse, J., & Minichino, J. (2020). Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning. Packt Publishing Ltd.
- Kinser, J. M. (2019). Image Operators: Image Processing in Python. CRC Press.