7 - 8.8. Generic Transformations

October 7, 2024

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

1 Introduction

`Generic Transformation` in image processing is a process of transforming an image from one domain to another domain. The transformation can be linear or non-linear. The transformation is done by applying a function to each pixel in the image. The function can be a simple mathematical function or a complex mathematical function. The transformation can be done in spatial domain or frequency domain. The transformation can be done in 2D or 3D. The transformation can be done in color domain or gray scale.

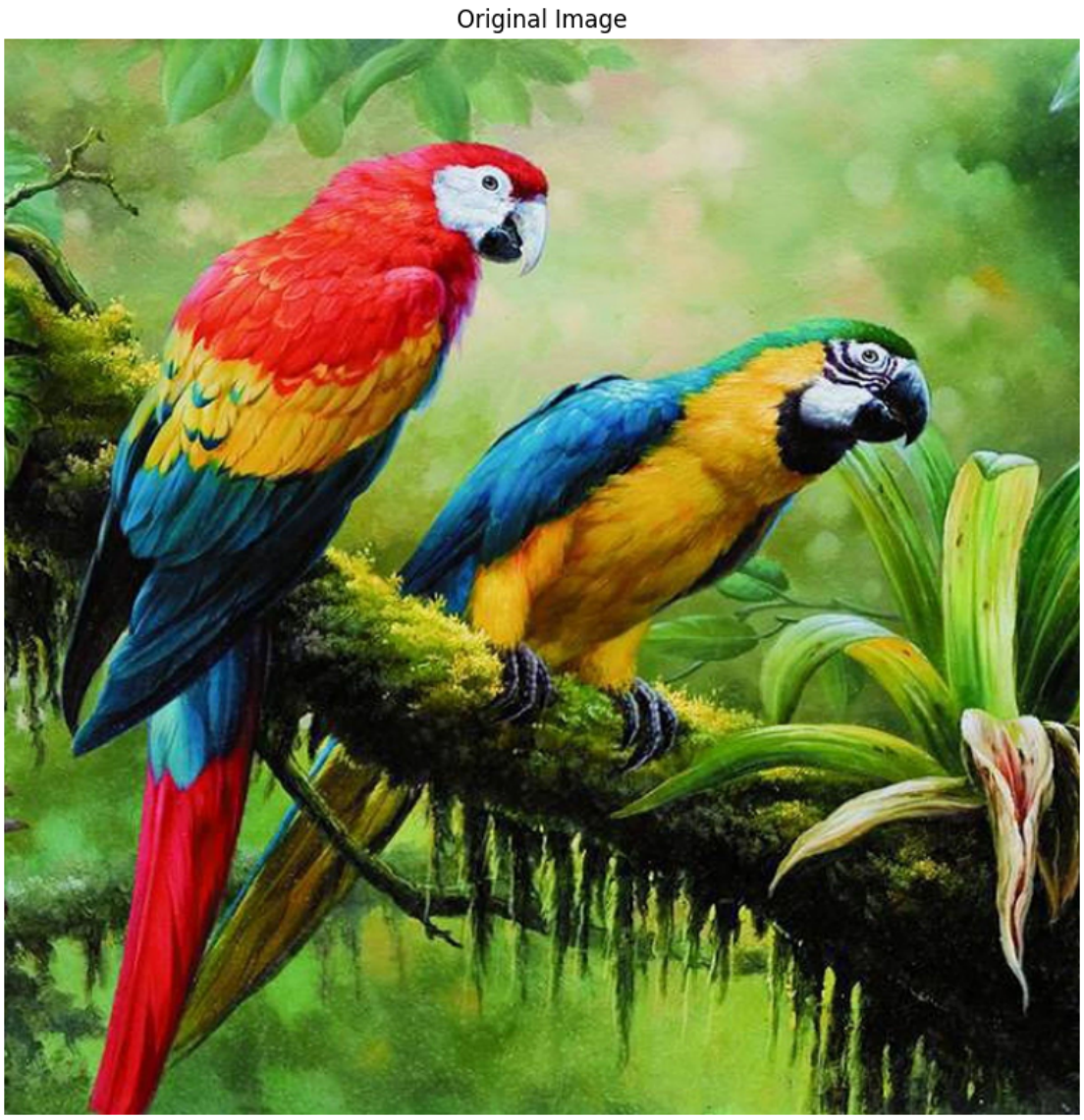
2 Setup

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

3 Initial Setup

[63]:
```
# 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', 'parrot.jpg')

# Read Image and convert to RGB
input_image = cv2.resize(cv2.cvtColor(cv2.imread(image_path), cv2.
 ↪COLOR_BGR2RGB), (720, 720))
```

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

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

plt.show()
```



4 Generic Transformation in Images

The `scipy.ndimage.geometric_transform` function is used to apply a generic transformation to an image. The function takes an input image and a function as input and returns the transformed image. The function is user defined and can be any mathematical function. The function takes the output coordinates as input and returns the input coordinates. The function can be a simple mathematical function or a complex mathematical function. The function can be a linear function or a non-linear function.

[64]:
```
# Geo Function
def geo_fun(outcoord, alpha=5, beta=5):
    num = len(outcoord)

    a = alpha * np.cos(outcoord[0]/beta) + outcoord[0]
    b = alpha * np.cos(outcoord[1]/beta) + outcoord[1]

    if num == 2:
        return (a, b)
    elif num == 3:
        return (a, b, outcoord[2])

# Flip Image
def flip_image(outcoord, size, axis=0):
    num = len(outcoord)

    a = outcoord[0]
    b = outcoord[1]

    def flip_x():
        return size[1] - outcoord[1] - 1

    def flip_y():
        return size[0] - outcoord[0] - 1

    if axis == 0:
        b = flip_x()
    elif axis == 1:
        a = flip_y()

    if num == 2:
        return (a, b)
    elif num == 3:
        return (a, b, outcoord[2])

# Generic Transformation
generic_geo_fun_image = nd.geometric_transform(input_image, geo_fun,␣
 ↪extra_keywords={'alpha': 5, 'beta': 5})
```

```
generic_flip_image_x = nd.geometric_transform(input_image, flip_image,␣
 ↪extra_keywords={'size': input_image.shape, 'axis': 0})
generic_flip_image_y = nd.geometric_transform(input_image, flip_image,␣
 ↪extra_keywords={'size': input_image.shape, 'axis': 1})

# Display Images
plt.figure("Generic Transformation", figsize=(20, 20))

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

plt.subplot(2, 2, 2)
plt.imshow(generic_geo_fun_image)
plt.title("Generic Transformation(geo_fun)")
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(generic_flip_image_x)
plt.title("Generic Transformation(flip_image(x))")
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(generic_flip_image_y)
plt.title("Generic Transformation(flip_image(y))")
plt.axis('off')

plt.show()
```

The code above shows the implementation of generic transformation in images using the `scipy.ndimage.geometric_transform` function. The code defines two functions `geo_fun` and `flip_image` which are used to transform the input image. The `geo_fun` function applies a non-linear transformation to the input image. The `flip_image` function flips the input image along the x-axis or y-axis. The `nd.geometric_transform` function is used to apply the transformation to the input image.

5 Summary

- `Generic Transformation` in image processing is a process of transforming an image from one domain to another domain.
- The transformation can be linear or non-linear.
- The transformation is done by applying a function to each pixel in the image.

- The function can be a simple mathematical function or a complex mathematical function.
- The transformation can be done in spatial domain or frequency domain.

Read More: - Geometric Transformation

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.