Geometric Transformations and Thresholding of Images using Opencv-Python

R. Devi
Assistant Professor
Department of Electronic and Communication Engineering
Saranathan College of Engineering,Trichy, India

Abstract

A geometric change is any bijection of a set having some geometric structure to itself or another such set. Specifically, a geometric change is a capacity whose space and range are sets of focuses. In this paper to apply diverse geometric change to pictures like interpretation, turn, relative change utilizing opencv-Python is presented. Geometric changes can be grouped by the measurement of their operand sets (along these lines recognizing planar changes and those of space, for instance). They can likewise be ordered by the properties they save. Displacements preserve distances and oriented angles; Iso-metries preserve angles and distances; Similarities preserve angles and ratios between distances; affine transformations preserve parallelism; projective transformations preserve collinearity.

Keywords- Transformations, Scaling, Translations, Affine

I. INTRODUCTION

In computer graphics and digital imaging, image scaling refers to the resizing of a digital image. In video technology, the magnification of digital material is known as up-scaling or resolution enhancement. When scaling a vector graphic image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of image quality. When scaling a raster graphics image, a new image with a higher or lower number of pixels must be generated. In the case of decreasing the pixel number (scaling down) this usually results in a visible quality loss. From the standpoint of digital signal processing, the scaling of raster graphics is a two-dimensional example of sample-rate conversion, the conversion of a discrete signal from a sampling rate (in this case the local sampling rate) to another.

The translate operator performs a geometric transformation which maps the position of each picture element in an input image into a new position in an output image, where the dimensionality of the two images often is, but need not necessarily be, the same. Under translation, an image element located at in the original is shifted to a new position in the corresponding output image by displacing it through a user-specified translation. The treatment of elements near image edges varies with implementation. Translation is used to improve visualization of an image, but also has a role as a preprocessor in applications where registration of two or more images is required. Translation is a special case of affine transformation. The rotation operator performs a geometric transform which maps the position of a picture element in an input image onto a position in an output image by rotating it through a user-specified angle about an origin. In most implementations, output locations which are outside the boundary of the image are ignored. Rotation is most commonly used to improve the visual appearance of an image, although it can be useful as a preprocessor in applications where directional operators are involved. Rotation is a special case of affine transformation.

II. LITERATURE SURVEY

Writing Survey In recent years, the recognition of Objects continuously and Image preparing has turned into a dynamic region of research and a few new methodologies have been proposed. A few scientists have directed many examinations about Object discovery 1. S.V. Viraktamath, Mukund Katti, Aditya Khatawkar and Pavan Kulkarni has led an investigation of openCV and furthermore have distributed an IEEE paper for Face Detection and Tracking utilizing OpenCV. Their work is connected with changing over web cam caught 2D Images and change over them into 3D Images identified with human faces by building 3D Geometry information yields [1]. 2. Ashish Pant, Arjun Arora, Sunnet Kumar and Prof. R.P. Arora shape DIT Dehradun have looked into about Image Processing and scrambling an Image with a specific end goal to exchange securely finished the systems. 3. They entitled their work as Sophisticated Image Encryption Using OpenCV [2]. 4. Kevinhughes, a first class individual in Opencv region has composed various sites containing ventures instructional exercises around there and ventures for introducing different virtual products [3]. 5. Serge Belongie and Jitendra Malik, individuals from IEEE have done a wast contemplate in the field of Shape Matching and Object Matching Based on their shapes, separating two question in light of the distinction in their shapes. 6. Orlando J. Tobias, and Rui Seara, Member, IEEE, have put their awesome endeavors studding the ways and strategies for Image Segmentation and histogram Thresholding.

III. TRANSFORMATION OF IMAGES

OpenCV gives two change capacities, cv2.warpAffine and cv2.warpPerspective, with which you can have a wide range of changes. cv2.warpAffine takes a 2x3 change grid while cv2.warpPerspective takes a 3x3 change framework as information. Scaling is simply resizing of the picture. OpenCV accompanies a capacity cv2.resize() for this reason. The extent of the picture can be indicated physically, or you can determine the scaling factor. Diverse addition strategies are utilized. Best introduction techniques are cv2.INTER_AREA for contracting and cv2.INTER_CUBIC (moderate) and cv2.INTER_LINEAR for zooming. As a matter of course, insertion strategy utilized is cv2.INTER_LINEAR for all resizing purposes. Interpretation is the moving of question's area. On the off chance that you know the move in (x,y) course, let it be , you can make the change grid as takes after

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

You can take make it into a Numpy array of type np.float32 and pass it into cv2.warpAffine() function. This figure 1.1 Shows simulation results for scaling of image using opencv-python.

Fig. 1.1: Scaling of Image

IV. IMAGE ROTATION AND AFFINE TRANSFORMATION USING OPENCV-PYTHON

Rotation of an image for an angle is achieved by the transformation matrix of the form

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

But OpenCV provides scaled rotation with adjustable center of rotation so that you can rotate at any location you prefer. Modified transformation matrix is given by

$$\begin{bmatrix} \alpha & \beta & (1-\alpha)\cdot center.x - \beta \cdot center.y \\ -\beta & \alpha & \beta \cdot center.x + (1-\alpha)\cdot center.y \end{bmatrix}$$

Where,

$$\alpha = scale \cdot \cos\theta,$$
$$\beta = scale \cdot \sin\theta$$

To find this transformation matrix, OpenCV provides a function, cv2.getRotationMatrix2D.

Fig. 1.2: Rotation of Images

V. AFFINE TRANSFORMATION & PRESPECTIVE TRANSFORMATION OF IMAGES USING OPENCV

In affine transformation, all parallel lines in the original image will still be parallel in the output image. To find the transformation matrix, we need three points from input image and their corresponding locations in output image. Then cv2.getAffineTransform will create a 2x3 matrix which is to be passed to cv2.warpAffine.

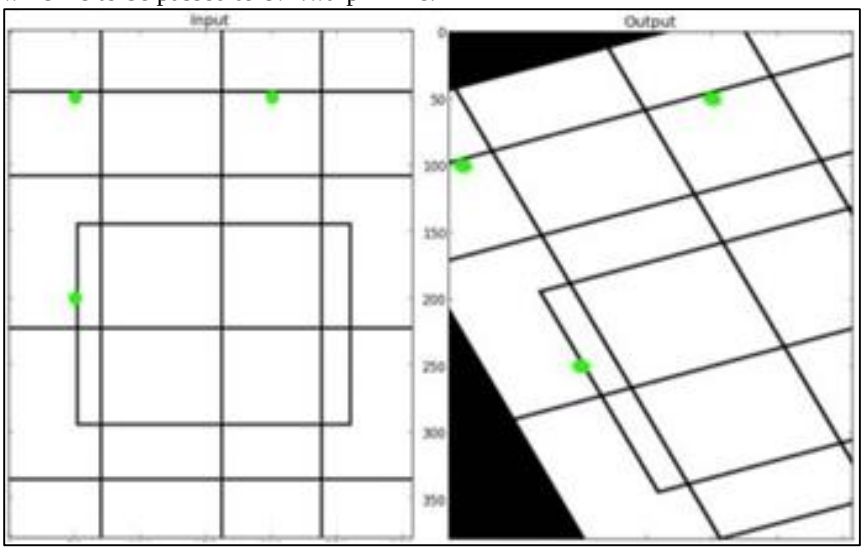

Fig. 1.3: Affine Transformation of Images

For perspective transformation, you need a 3x3 transformation matrix. Straight lines will remain straight even after the transformation. To find this transformation matrix, you need 4 points on the input image and corresponding points on the output image. Among these 4 points, 3 of them should not be collinear.

Then transformation matrix can be found by the function cv2.getPerspectiveTransform.

Then apply cv2.warpPerspective with this 3x3 transformation matrix.

VI. IMAGE THRESHOLDING USING OPENCV-PYTHON

The function used is cv2.threshold. First argument is the source image, which should be a grayscale image. Second argument is the threshold value which is used to classify the pixel values. Third argument is the maxVal which represents the value to be given if pixel value is more than (sometimes less than) the threshold value. OpenCV provides different styles of thresholding and it is decided by the fourth parameter of the function. Different types are:

- cv2.THRESH_BINARY
- cv2.THRESH_BINARY_INV
- cv2.THRESH_TRUNC
- cv2.THRESH_TOZERO
- cv2.THRESH_TOZERO_INV

In the previous section, we used a global value as threshold value. But it may not be good in all the conditions where image has different lighting conditions in different areas. In that case, we go for adaptive thresholding. In this, the algorithm calculate the threshold for a small regions of the image. So we get different thresholds for different regions of the same image and it gives us better results for images with varying illumination.

It has three 'special' input parameters and only one output argument.

Adaptive Method - It decides how thresholding value is calculated.

cv2.ADAPTIVE_THRESH_MEAN_C:Threshold value is the mean of neighbourhood area.

cv2.ADAPTIVE_THRESH_GAUSSIAN_C : threshold value is the weighted sum of neighbourhood values where weights are a gaussian window.

Block Size - It decides the size of neighbourhood area.

C - It is just a constant which is subtracted from the mean or weighted mean calculated

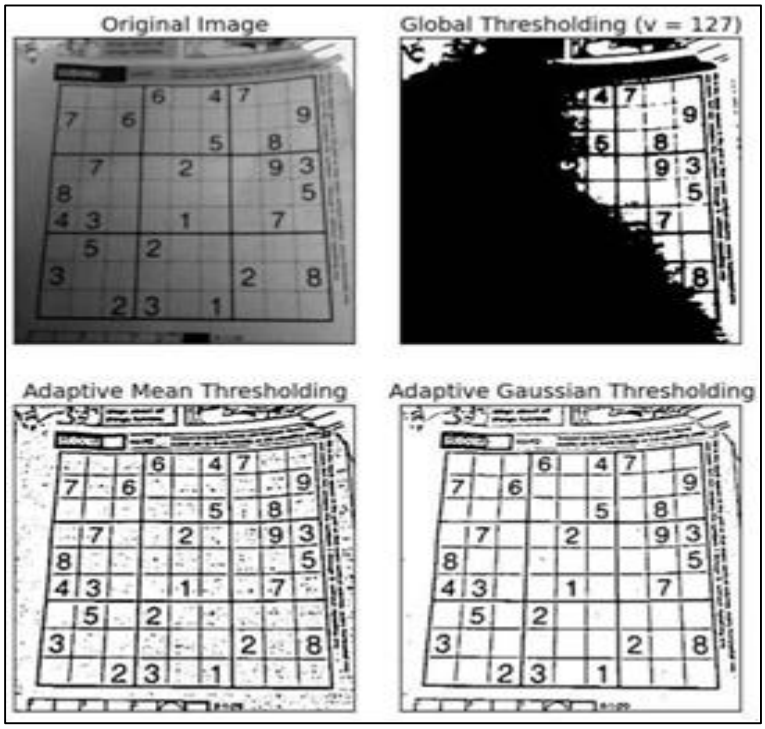

Fig. 1.4: Adaptive Thresholding

VII. CONCLUSION AND FUTURE WORK

Image Thresholding and geomentric transformation of images is done by using opencv –python. Geomrntric transformations of images is done by using scaling, rotation, transformations of images is presented here. Image thresholding is done by using adaptive thresholding.This is used to reduce the noise of images and increases the contrast of the images.

REFERENCES

[1] Viraktamath SV, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni, "Face Detection and Tracking using OpenCV," The SIJ Transaction on Computer Networks & Communication Engineering (CNCE), 2013, 1(3).
[2] Pant A, Arora A, Kumar S, Arora RP. "Sophisticated Image Encryption Using OpenCV," International Journal of Advances Research in Computer Science and Software Engineering, 2012, 2(1).
[3] Kevin Hughes – One more robot learn to see (http://kevinhughes.ca)
[4] Belongie S, Malik J, Puzicha J. "Shape Matching and Object Recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002; 24(4):509-522,
[5] Tobias OJ, Seara R. "Image Segmentation by Histogram Thresholding Using Fuzzy Sets," IEEE Transactions on Image Processing, 2002; 11(12):1457-1465.
[6] http://www.opencv.org