International Journal of Engineering & Technology, 7 (4.19) (2018) 50-57

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Analysis of Iterated Affine Transformation Function and Linear Mapping for Content Preservation

Ankit Garg[1*], Ashish Negi[2], Geeta Chauhan[3]

[1]*Research Scholar, Uttarakhand Technical University, Dehradun, India.*
[2]*Professor & Head, GovindBallabh Pant Engineering College, PauriGarhwal, India*
[3]*Deputy Director, Mahadevi Institute of Technology, Dehradun, UK, India*
**Corresponding Author Email: [1]ankitgitm@gmail.com*

Abstract

In image scaling contents of image can be distorted which are required to preserve using linear mapping. Geometric transformations can preserve structural properties i.e. parallelism, colinearity and orientation. It is highly desirable to preserve structural properties of image contents because human visual system is very sensitive to distortion of objects. In this paper image scaling is performed using iterative affine transformation and results show that linear mapping function applied on affine space preserve affine properties under affine transformation. A number of scaling operations are applied on image using iterative affine transformation and for each iteration linear mapping is performed to preserved object structure. Analysis present in this paper shows that in image scaling preservation of image content is possible under iterative affine transformation and linear mapping. Image artifacts can be minimize using saliency based antialiasing algorithm after affine transformation.

Keywords: *Affine Transformation; Aliasing; Face Recognition; Fractal; Iterated function system; Resampling.*

1. Introduction

In computer graphics affine transformation preserves affine property of affine space i.e. distortion of line, point and planes. Affine space generalizes the properties of Euclidian geometry in such a way that it does not include the concept of distance, angle, area and preservation of distance between points but preserve following properties:

- Parallelism- Parallel lines remain parallel.
- Co linearity- The line segments lying on the same line remain collinear.
- Ratio of Distance- Preserve ration of distance.
- Orientation- Preserve orientation of line segments.
- Convexity of sets- A convex set continues to be convex after the transformation.

Geometric transformation which can be included in affine transformation are translation, rotation, scaling, reflection, shearing and similarity transformation and any composite transformation which is combination of all the 2D transformation. Affine transformation is useful in various areas of computer science. T. Gangopadhyay [1] present study on the effect of multiple rotations on affine transformations with related trigonometric coefficients in terms of the iterated function system fractals generated by them by using unified set of equations. DandanZuo [2] proposed algorithm which utilize affine map to repair the structure of traffic signs in the real images. N. Paniphi [3] discussed various aspects related to the geometric errors observed in a satellite image and polygonal affine transformation to remove them. P. K. Mohanty [4] proposed an algorithm for designing face recognition algorithms based solely on affine transforms. C.V. Priscilla [5] proposed a method for image registration and noise reduction using affine transformation. M. Toorani [6] introduces a secure and efficient symmetric cryptosystem based on affine transformation. Shen Wang [7] proposed a lossless JPEG2000 image steganography algorithm based on affine transformation. H. Chen [8] proposed a novel method based on affine transformation, which can correct and stitch the doppler beam sharpening image effectively. In image retargeting, images are displayed on different screen size with different aspect ratio. For preserving highly noticeable content in image researchers have found various methods. C. Chang [9] proposed an image resizing method in content aware fashion which simultaneously preserves both salient image features and line structure properties. Proposed method combines mesh deformations which control content preservation and for image resizing with similarity transforms for line features.

In Figure 4 image depicts scaling operation and linear mapping for content preservation. Output of transformation shows image artifacts which must be avoided through precise antialiasing algorithm. In this paper affine transformation is analyzed with Iterated function system to maintain affine property.Yao Xiao [13] proposed fast skewed object matching with adaptive rotation steps based on affine transformation. Lei Yu [14] proposed algorithm which focused on problem associated with imprecise location and trouble on the conventional large-screen, using projectors and Kinect with affine transformation algorithm to get exact coordinates interaction area and capture interactive action. RediaRedzuwan [15] proposed a method to for correcting position by affine matrix transformation based on image features detection, description and matching and affine transformation estimation.Xuan Li [16] proposed novel regression strategy by introducing affine transformation

In this paper, Section (2) represents utility of forward and backward mapping methods for calculating corresponding relative coordinate positions or target object in image. Section (3, 4) pre-

sents role contractive affine transformation to generate fractal images. In section (5) algorithm for iterated affine transformation is presented. Section (6) present result of iterated affine transformation and linear mapping for content preservation of fractal images. Section (7) present effect of iterated affine transformation over image to maintain affine properties. To analyze affine map, affine properties are justified using forward and backward linear mapping function. Section (8) represents problem of artifacts after transformation which can be removed by antialiasing algorithm [17] which is based on saliency map.

2. Mapping Procedure

2.1 Forward Method

In Forward Mapping each pixel of source image is mapped with the pixel in target image [10].
In Figure 1,forward mapping function f(iu,iv) is applied on pixel (iu, iv) in source image to map it with corresponding pixel (x, y) in target image. Buffer is used to store pixel location of source image to find out precise locations of pixels after forward mapping. High computation time and speed is required to implement algorithm which incorporates this method.

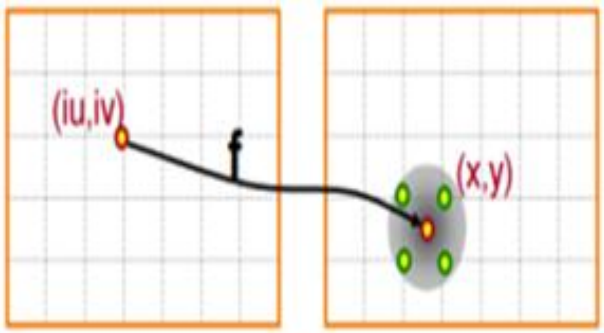


Fig. 1: (a) Source Image (b) Target Image

2.2 Reversed Method

To map image pixels a inverse transformation function is used which maps target image pixel locations on source image. Pixel of the destination image (ix, iy) is mapped to the backward function f to the pixel (u, v) in the source image [10].

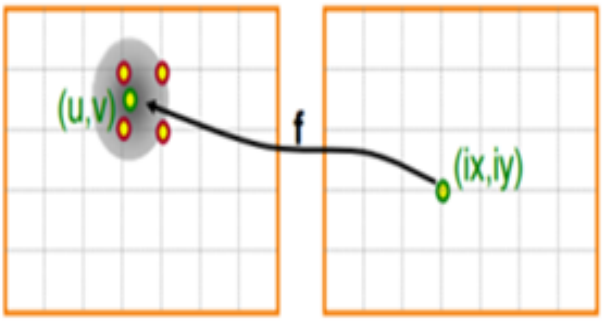


Fig. 2: (a) Target Image (b) Source Image

3. Affine Transformation and Image Mapping

Fractals can be generated using iterated function system [11] using following steps.

- Implement combined transformations.
- Draw any object.
- Applying geometric affine transformations on the initial object.
- Iteratively apply transformation on initial object to get new transformed object.
- Apply image mapping function, recursively to map coordinate of source fractal image to target image.
- Repeat step 5, until attractor will not generated.

4. Contractive Affine Transformation

A contraction mapping is a transformation where any two distinct points are brought closer together by the transformation.
A contracting affine transformation is an affine transformation that is also a contraction mapping. Contraction mappings have a very important property - they leave exactly one point fixed [12].
Following combine transformation matrix can applied on the image.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \quad (1)$$

$$= s \begin{bmatrix} cos\alpha(x - x_0) + & sin\alpha(y - y_0) \\ -sin\alpha(x - x_0) + & cos\alpha(y - y_0) \end{bmatrix} \quad (2)$$

.
Separating the equations,

$$x' = (s\, cos\alpha)x + (s\, sin\alpha)y - s(x_0\, cos\alpha + y_0\, sin\alpha) \quad (3)$$

$$y' = (-s\, sin\alpha)x + (s\, cos\alpha)y + s(x_0\, sin\alpha - y_0\, cos\alpha) \quad (4)$$

This can be also written as

$$x' = ax - by + c \quad (5)$$

$$y' = bx + ay + d \quad (6)$$

Where

$$a = s\, cos\alpha \quad (7)$$

$$b = -\, sin\alpha \quad (8)$$

The scale factor s is then defined by

$$s = \sqrt{a^2 + b^2} \quad (9)$$

and the rotation angle by

$$\alpha = tan^{-1}\left(\frac{b}{a}\right) \quad (10)$$

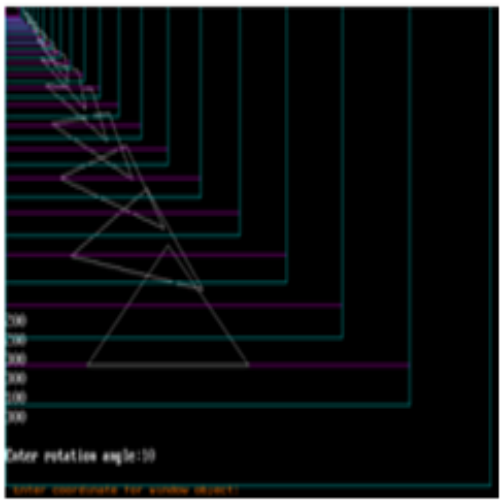
An affine transformation of $\mathbb{R}^n$is a map$F: \mathbb{R}^n \to \mathbb{R}^n$of the form

$$F(p) = Ap + q \quad (11)$$

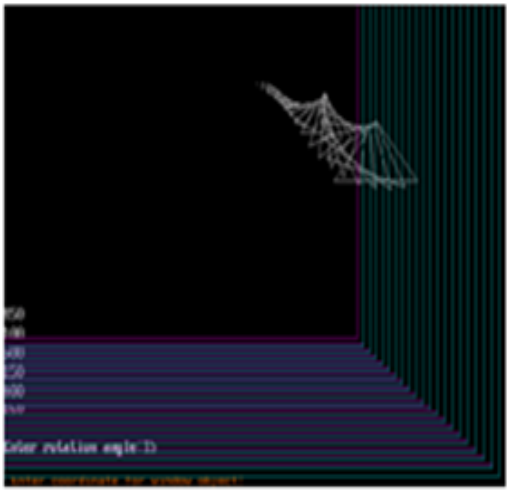
for all $\rho \in \mathbb{R}^n$, where A is a linear transformation of $\mathbb{R}^n$.
If $\det(a) > 0$, the transformation is orientation-preserving;
if,$\det(a) < 0$ is orientation-reversing.

5. Iterative Affine Transformation and Linear Mapping

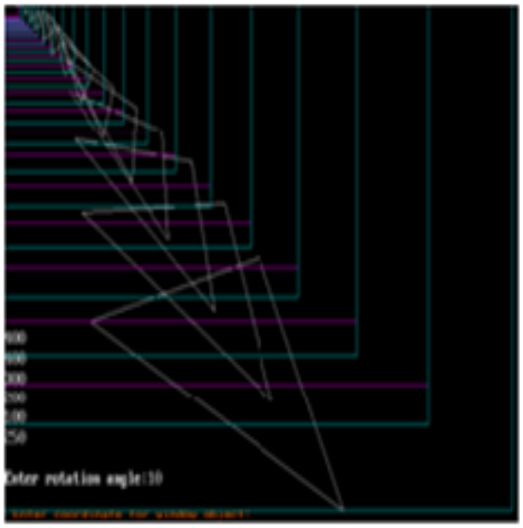
In this section affine transformation is implemented and analyzed affine properties. Iterated function system is also implemented which combines for Iterative affine transformation to produce image and iterative forward and reverse image mapping method calculates relative position of each pixel of source image in to target image. After each iteration of implemented iterated function affine properties are analyzed for geometric distortion of object and their preservation.



(a)



(b)

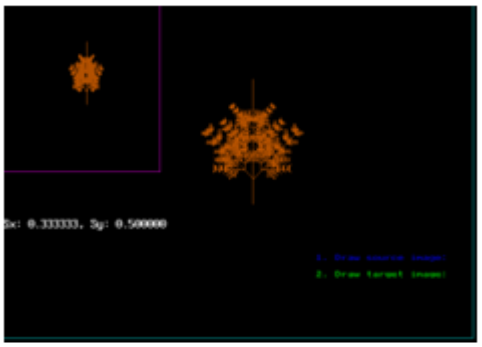


(c)

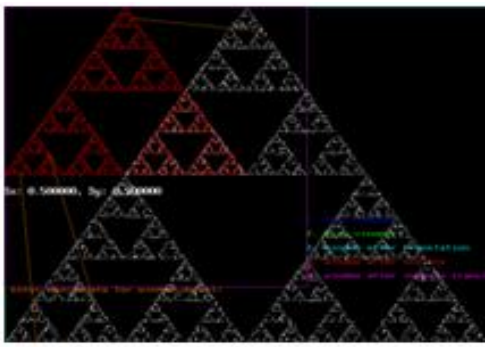
Fig. 3: (a) Reducing image with scaling factors Sx=0.5, Sy=0.5, rotation angle 10^0, (b) scaling factors Sx=0.5, Sy=0.5, rotation angle 10^0, (c) Scaling factorsSx=0.3, Sy=0.5 and rotation angle 15^0.

Table 1: Constructing Affine Transformation

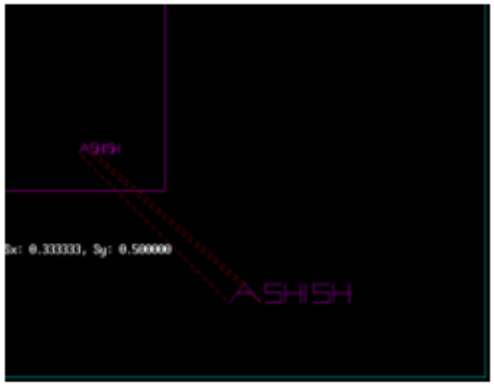
Figure	Matrix				Translation	
	a	B	c	d	E	f
Figure 4(a)	0.2	0.231	0.234	0.2	-1	-2
Figure 4(b)	0.3	0.32	0.273	0.4	-1``	-2
Figure 4(c)	0.4	0.212	0.221	0.3	-1	-0.5



(a)



(b)



(c)

(d)

Fig. 4: (a) Affine transformation with scaling factorsSx=0.5 &Sy=0.5, (b) Scaling factorsSx=0.3 and Sy=0.5, (c) Scaling factors Sx=0.5 and Sy=0.5, (d) Scaling factorsSx=0.5 and Sy=0.5 with mapping, (d) Scaling factorsSx=0.3 and Sy=0.5 with image mapping.

In figure 4 (c) it is observed that affine transformation map preserves the property of parallelism, Co linearity and ratio of distance.

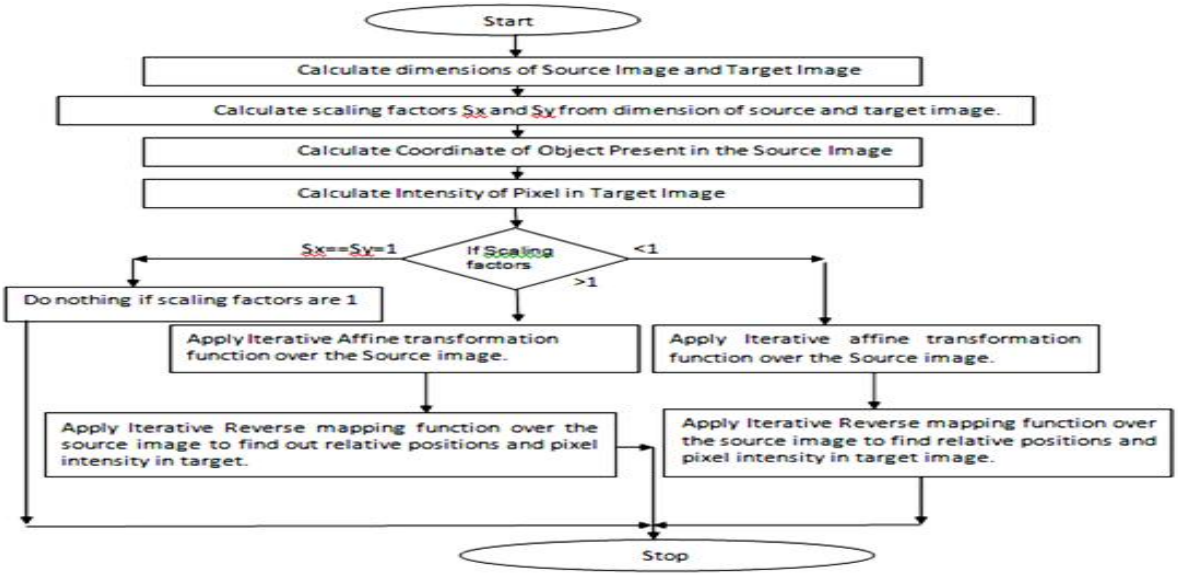


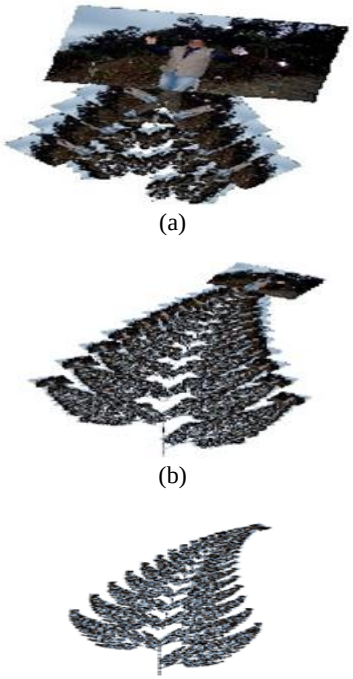
Fig. 5: Flowchart of affine transformation over image to reduce and expand size

Table 2: Iterated Affine transformation Function applied on image, For Figure 3(a)

Iteration	Dimension of Source Image								Dimension of Target Image							
	X1	Y1	X2	Y2	X3	Y3	X4	Y4	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	0	0	600	0	600	600	0	600	0	0	500	0	500	500	0	500
2	0	0	500	0	500	500	0	500	0	0	417	0	417	417	0	417
3	0	0	417	0	417	417	0	417	0	0	347	0	347	347	0	347
4	0	0	347	0	347	347	0	347	0	0	289	0	289	289	0	289
5	0	0	289	0	289	289	0	289	0	0	241	0	241	241	0	241
6	0	0	241	0	241	241	0	241	0	0	201	0	201	201	0	201
7	0	0	201	0	201	201	0	201	0	0	167	0	167	167	0	167
8	0	0	167	0	167	167	0	167	0	0	140	0	140	140	0	140
9	0	0	140	0	140	140	0	140	0	0	116	0	116	116	0	116
10	0	0	116	0	116	116	0	116	0	0	97	0	97	97	0	97
11	0	0	97	0	97	97	0	97	0	0	81	0	81	81	0	81
12	0	0	81	0	81	81	0	81	0	0	67	0	67	67	0	67
13	0	0	67	0	67	67	0	67	0	0	56	0	56	56	0	56
14	0	0	56	0	56	56	0	56	0	0	47	0	47	47	0	47
15	0	0	47	0	47	47	0	47	0	0	39	0	39	39	0	39
16	0	0	39	0	39	39	0	39	0	0	32	0	32	32	0	32
17	0	0	32	0	32	32	0	32	0	0	27	0	27	27	0	27
18	0	0	27	0	27	27	0	27	0	0	23	0	23	23	0	23
19	0	0	23	0	23	23	0	23	0	0	19	0	19	19	0	19
20	0	0	19	0	19	19	0	19	0	0	16	0	16	16	0	16

Table 3: Forward and Backward Image Mapping For Figure 4 (a)

Iteration	Pixel Locations in Source Image						Pixel Location in Target Image After Mapping						Xc	Yc
	X1	Y1	X2	Y2	X3	Y3	X1	Y1	X2	Y2	X3	Y3		
1	200	0	300	300	100	300	222	30	215	260	60	207	166	166
2	222	30	215	260	60	207	218	59	148	214	43	137	137	137
3	218	59	148	214	43	137	198	82	101	169	40	89	113	113
4	198	82	101	169	40	89	169	95	69	128	44	59	94	94
5	169	95	69	128	44	59	134	98	49	94	49	42	77	78
6	134	98	49	94	49	42	101	92	37	67	52	33	63	64
7	101	92	37	67	52	33	73	80	31	47	52	29	52	52
8	73	80	31	47	52	29	50	67	28	35	48	28	42	43
9	50	67	28	35	48	28	34	52	26	27	42	27	34	35
10	34	52	26	27	42	27	24	40	24	22	36	26	28	29
11	24	40	24	22	36	26	18	30	23	19	29	24	23	24
12	18	30	23	19	29	24	14	22	20	16	22	21	19	20
13	14	22	20	16	22	21	11	16	16	14	17	17	15	16
14	11	16	16	14	17	17	10	12	13	12	13	13	12	12
15	10	12	13	12	13	13	9	10	9	10	9	10	9	10
16	9	10	9	10	9	10	7	8	7	8	7	8	7	8
17	7	8	7	8	7	8	5	6	5	6	5	6	5	6
18	5	6	5	6	5	6	4	5	4	5	4	5	4	5
19	4	5	4	5	4	5	3	4	3	4	3	4	3	4
20	3	4	3	4	3	4	2	3	2	3	2	3	2	3



(a) (b) (c)

Fig. 6: (a) Iterated Function System after applying 4 Iteration, (b) Fractal Image after 14 Iterations, (c) Fractal Image after 20 Iterations

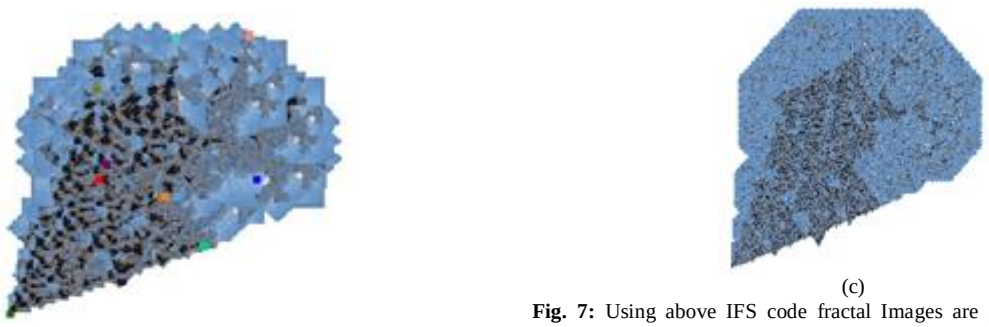
Table 4: Iterated Function System Code Using Matrix Compact Form for Fern Generation with 4 Iterations

Iterations	Matrix				Translation		Probability
	a	B	c	d	E	f	p
Iteration 1	0.00	0.00	0.00	0.160	0.00	0.00	0.010
Iteration 2	0.850	0.040	-0.040	0.850	0.00	1.600	0.850
Iteration 3	0.200	-0.260	0.230	0.220	0.00	1.600	0.070
Iteration 4	-0.150	0.280	0.260	0.240	0.00	0.440	0.070

Above 4 Iterations are applied on image with Iterated function coefficient to generate fractal images. Geometric transformation which are including in iterations are translation, rotation, scaling. When scaling transformation is applied iteratively on image line structure properties can be preserved.

Table 5: Code Using Scale/Rotate Form with 10 Iterations

Iterations	Scale		Rotation			Translation	Probability
Iteration 1	0.707	0.707	45.00	45.00	0.00	0.00	0.167
Iteration 2	0.707	0.707	- 45.00	- 45.00	0.500	0.500	0.167
Iteration 3	0.500	0.500	0.00	0.00	- 0.283	- 0.444	0.083
Iteration 4	0.500	0.500	0.00	0.00	.024	.054	0.083
Iteration 5	0.500	0.500	0.00	0.00	.204	-0.057	0.083
Iteration 6	0.500	0.500	0.00	0.00	0.336	-0.219	0.083
Iteration 7	0.500	0.500	0.00	0.00	0.244	0.464	0.083
Iteration 8	0.500	0.500	0.00	0.00	-0.007	0.297	0.083
Iteration 9	0.500	0.500	0.00	0.00	0.477	0.475	0.083
Iteration 10	0.500	0.500	0.00	0.00	0.021	0.278	0.083



(a) (b) (c)

Fig. 7: Using above IFS code fractal Images are generated (a) Fractal Image Using 40 Iterations with 10 Fixed Points, (b) Fractal image using 100 Iterations, (c) Fractal Image using 200 Iterations

6. Analysis and Results

Let (wxmin, wymin) and (wxmax, wymax) be the corners of target image and (vxmin, vymin) and (vxmax, vymax) be the coordinates of source image. Let assume that aspect ratio of both source image and target image is same so,

$$sx = \frac{wx2 - wx1}{vx2 - vx1}, sy = \frac{wy2 - wy1}{vy2 - vy1}$$

- **Property 1 (Co linearity)**-Three or more points which lie on the same line (called collinear points) continue to be collinear after the transformation.

Proof-
Let three points (a1,b1), (a2,b2), (a3,b3) be collinear.

$$\frac{a2-a1}{b2-b1}=\frac{a3-a2}{b3-b2}=\frac{a3-a1}{b3-b1} \quad \dots (12)$$

Let the coordinates of three points in view port be (a1`,b1`), (a2`,b2`), and (a3`,b3`).

$$a1' = vxmin + (a1 - wxmin)s_x$$
$$b1' = vymin + (b1 - wymin)s_y$$
$$a2' = vxmin + (a2 - wxmin)s_x$$
$$b2' = vymin + (b2 - wymin)s_y$$
$$a3' = vxmin + (a3 - wxmin)s_x$$
$$b3' = vymin + (b3 - wymin)s_y$$
$$m1 = \frac{a2' - a1'}{b2' - b1'}$$

$$= \frac{\{vxmin + (a2 - wxmin)s_x - vxmin + (a1 - wxmin)s_x\}}{\{vymin + (b2 - wymin)s_y - vymin + (b1 - wymin)s_y\}} \quad (13)$$

$$\frac{(a2 - a1)s_x}{(b2 - b1)s_y}$$
$$m2 = \frac{a3' - a1'}{b3' - b1'}$$

$$= \frac{\{vxmin + (a3 - wxmin)s_x - vxmin + (a1 - wxmin)s_x\}}{\{vymin + (b3 - wymin)s_y - vymin + (b1 - wymin)s_y\}} \quad (14)$$

$$\frac{(a3-a1)s_x}{(b3-b1)s_y} \quad (15)$$

Substituting value of $\frac{(a3-a1)}{(b3-b1)}$ from eq. (12) in eq. (15) we get

$$m2 = \frac{(a2 - a1)s_x}{(b2 - b1)s_y}$$
$$\frac{a2 - a1}{b2 - b1} = \frac{a4 - a3}{b4 - b1}$$

m1=m2

Points are collinear since one point is common and slopes are equal.

- **Property 2 (Parallelism) -**Two or more lines which are parallel, continue to be parallel after the transformation.

Proof- Let AB and CD be two lines with coordinates (x1,y1) (x2,y2) and (x3,y3) (x4,y4) respectively be parallel. The slope of AB and CD will be equal since both the lines are equal.

$$\frac{a2-a1}{b2-b1}=\frac{a4-a3}{b4-b1} \quad (16)$$

$$a1' = vxmin + (a1 - wxmin)s_x$$
$$b1' = vymin + (b1 - wymin)s_y$$
$$a2' = vxmin + (a2 - wxmin)s_x$$
$$b2' = vymin + (b2 - wymin)s_y$$
$$a3' = vxmin + (a3 - wxmin)s_x$$
$$b3' = vymin + (b3 - wymin)s_y$$
$$a4' = vxmin + (a4 - wxmin)s_x$$
$$b4' = vymin + (b4 - wymin)s_y$$
$$m2 = \frac{a4' - a3'}{b4 - b3'}$$

$$= \frac{\{vxmin + (a4 - wxmin)s_x - vxmin + (a3 - wxmin)s_x\}}{\{vymin + (b4 - wymin)s_y - vymin + (b3 - wymin)s_y\}} \quad (17)$$

$$\frac{(a4 - a3)s_x}{(b4 - b3)s_y}$$

Substituting value of a4-a3/b4-b3 from (16), we get

$$m2 = \frac{(a2-a1)s_x}{(b2-b1)s_y}$$

m1 = m2

Lines are parallel since there are no common points and slope is equal.

- **Property 3 (Ratio of Distance) -**Along a line for distinct collinear points the ratio of distance between them remains same after undergoing transformation.

Proof-Let three points (a1, b1), (a2, b2), (a3, b3) on line AB
Since the three points are collinear. Therefore slopes of line formed by considering any 2 points will be equal.

$$\frac{a2-a1}{b2-b1}=\frac{a3-a2}{b3-b2}=\frac{a3-a1}{b3-b1} \quad (18)$$

Let the coordinates of three points in target image be (a1`, b1`), (a2`, b2`), and (a3`, b3`). Calculate ratio of distance by substituting coordinate position in distance formula.
In target image Ratio of distance will be:

$$r1 = \frac{\sqrt{\{(a2'-a1')^2+(b2'-b1')^2\}}}{\{(a3'-a1')^2+(b3'-b1')^2\}} \quad (19)$$

$$r1 = \frac{\sqrt{\{((a2-a1)s_x)^2+((b2-b1)s_y)^2\}}}{\sqrt{\{((a3-a1)s_x)^2+((b3-b1)s_y)^2\}}} \quad (20)$$

Taking b2-b1 and b3-b1 as common from numerator and denominator respectively, we get

$$r1 = \frac{m\times\sqrt{\{(m\times sx)^2+s_y{}^2\}}}{\sqrt{\{(m\times s_x)^2+s_y{}^2\}}} \quad (21)$$

r1 = m
Ratio of distance between will be

$$r2 = \frac{\sqrt{\{(a2'-a1')^2+(b2'-b1')^2\}}}{\{(a3'-a2')^2+(b3'-b2')^2\}} \quad (22)$$

$$r2 = \frac{\sqrt{\{((a2-a1)s_x)^2+((b2-b1)s_y)^2\}}}{\sqrt{\{((a3-a2)s_x)^2+((b3-b2)s_y)^2\}}} \quad (23)$$

$$r2 = \frac{m\times\sqrt{\{(m\times sx)^2+s_y{}^2\}}}{\sqrt{\{(m\times s_x)^2+s_y{}^2\}}} \quad (24)$$

r2 = m
r1 = r2
Along a line for distinct collinear points the ratio of distance between them remains same after undergoing transformation.

- **Property 4 (Distance between Points)** – An affine transformation does not preserve distances between points.

Proof-
Let three points (a1,b1), (a2,b2), (a3,b3) makes triangle in source image and after relative mapping we get corresponding points (a1`,b1`), (a2`,b2`), and (a3`,b3`).
Distance between points in source image (a1, b1), (a2, b2) is

$$r1 = \sqrt{\{(a1 - a2)^2 + (b1 - b2)^2\}} \quad (25)$$

Distance between points in target image (a1`,b1`), (a2`,b2`)is

$$r2 = \sqrt{\{((a2 - a1)s_x)^2 + ((b2 - b1)s_y)^2\}} \quad (26)$$

$$r2 = \sqrt{\{(a2' - a1')^2 + (b2' - b1')^2\}} \quad (27)$$

Hence, r1≠r2.

➢ **Property 5 (Area of Object)** – An affine transformation does not preserve area of object.

Proof:

Let three points (a1,b1), (a2,b2), (a3,b3) makes right triangle in source image where distance between (a1, b1) and (a2, b2) gives length and distance between (a1, a2) and (a3, b3) gives height. After relative mapping we get corresponding points (a1`,b1`), (a2`,b2`), and (a3`,b3`).

Substitute relative coordinates in formula for calculating area of triangle.

Area of right angle in source image is

$$A1 = \frac{\sqrt{\{(a1 - a2)^2 + (b1 - b2)^2\}} \times \sqrt{\{(a1 - a2)^2 + (a3 - b3)^2\}}}{2} \quad (28)$$

Area of right triangle in target image is

$$A2 = \frac{\sqrt{\{(a1' - a2')^2 + (b1' - b2')^2\}} \times \sqrt{\{(a1' - a2')^2 + (a3' - b3')^2\}}}{2} \quad (29)$$

$$A2 = \frac{\sqrt{\{(a1 - a2)^2 + (b1 - b2)^2\}} \times \sqrt{\{(a1 - a2)^2 + (a3 - b3)^2\}}}{2} \quad (30)$$

Hence, A1≠A2

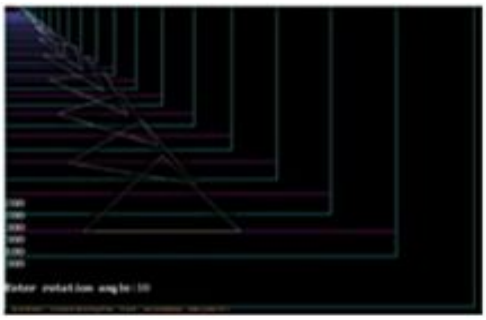
Table 6: Analysis of affine Properties after Iterative Affine Transformation

Properties of Affine Map	Object of Input Image (Before Affine Transformation)						Coordinates of Object After Linear Mapping						Outcome
	X1	Y1	X2	Y2	X3	Y3	X`1	Y`1	X`2	Y`2	X`3	Y`3	
Co linearity	20	40	30	50	40	60	5	7.5	6.25	8.75	7.5	10	Valid
Parallelism	0	0	4	4	8	8	3	1	7	5	11	9	Valid
Ratio of Distance	20	40	30	50	40	60	5	7.5	6.25	8.75	7.5	10	Valid
Angle or Orientation	20	40	30	50	40	60	5	7.5	6.25	8.75	7.5	10	Valid
Distance Between Points	20	40	30	50	40	60	5	7.5	6.25	8.75	7.5	10	Invalid
Area of Object	20	40	30	50	40	60	5	7.5	6.25	8.75	7.5	10	Invalid

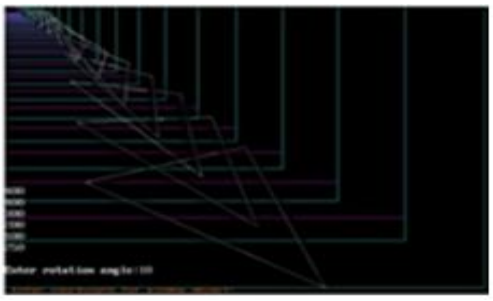
7. Antialiasing

Iterative affine transformation with image mapping produces aliasing effects due to change in angle of lines structures. A precise antialiasing algorithm is required to minimize image artifacts.

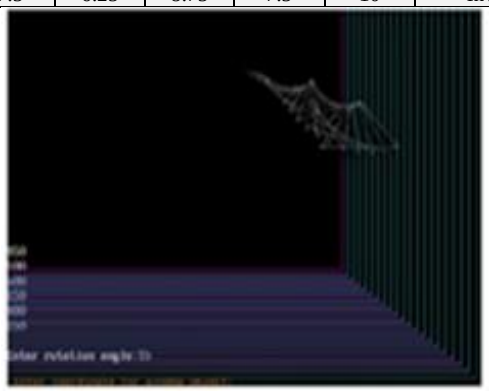
A saliency based antialiasing algorithm can be implemented with iterative affine transformation for smoothening staircase effects.



(a)



(b)



(c)

Fig. 8: Iterative affine transformation and image mapping with antialiasing

8. Discussion and Conclusion

Affine transformation is a linear mapping function between affine space which preserves affine properties. The affine transformation corrects geometric distortions or deformations of objects present in image that occur with non-ideal camera angles. For example, in content aware image resizing line structure properties can be distorted which can be preserve with the use of affine transformation and it is also applicable in satellite imagery, face recognition, stenography, image stitching and image registration. Image mapping functions can be used to find out relative position of image pixel in source image into target image.

This paper presents analysis of affine transformation to preserve line structure properties i.e. co linearity, parallelism, ratio of distance, and convex set. Iterative Image mapping procedure is also used to map coordinate of source image with relative coordinate of target image after affine transformation to maintain affine properties. Result of geometric affine transformation produce aliasing affects which can be rectifying by anti-aliasing algorithm. For

implementation of iterative affine transformation C language is used and for removal of aliasing problem anti aliasing filters are used over image using MATLAB.

References

[1] T. Gangopadhyay, "The Effect of Multiple Rotations on a Unified System of Affine Transformations with related Trigonometric Coefficients",International Journal of Computer Applications (0975 – 8887), Volume 139 – No.2, April 2016. pp. 30-35.

[2] DandanZuo, Xin Liu, and Shuangdong Zhu, "Application of Affine Transformation in Traffic Sign Detection", International Conference on Intelligent Control and Information Processing, 2010, pp. 277-280.

[3] Narayan Paniphi, "Image Registration using Polynomial Affie Transformation", Defence Science Journal, Vol. 52, No. 3, July 2002, pp. 253-259.

[4] Pranab K. Mohanty, Sudeep Sarkar, and RangacharKasturi,"Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005. pp. 1-8.

[5] C.VictoriaPriscilla,B.Poorna, "Image Registration and Nose Detection Using Affine Transformation", International .Journal of.Computer Technology & Applications, Vol 4 (2), pp. 209-216.

[6] Mohsen Toorani, AbolfazlFalahati, "A secure cryptosystem based on affine transformation",Proceedings of 2011 security and communication network. pp. 207-215.

[7] Shen Wang, Xianhua Song, XiamuNiu, "An Affine Transformation Based Image Steganography Approach", International Journal of Digital Content Technology and its Applications (JDCTA) Vol.6, No.1, January 2012.

[8] Hongmeng Chen, Ming Li, Yunlong Lu, Yan Wu, "A DBS Image Stitching Algorithm Based On Affine Transformation", Proceedings of Radar Conference, 2013.

[9] Che-Han Chang, Yung-Yu Chuang, "A Line-Structure-Preserving Approach to Image Resizing", Proceedings of Computer Vision and Pattern Recognition (CVPR), 2012. pp. 1075-1082.

[10] NafiseZarei, AbdolrezaSepyani, "Different methods of image mapping, its advantages and disadvantages", International Academic Journal of Science and Engineering Vol. 3, No. 4, 2016, pp. 1-10.

[11] R.Uthayakumar&G.ArockiaPrabakar, "Creation of Fractal Objects by Using Iterated Function System", Computing Communication & Networking Technologies (ICCCNT), 2012.

[12] Ljubiˇsa M. Koci´c and Marjan M. Mateji´c,"Contractive Affine Transformations Of Complex Plane & Applications", Facta Universities (NIS) ˇ Ser. Math. Inform. 21 (2006), pp. 65–75.

[13] Yao Xiao, "Fast skewed object matching with adaptive rotation steps based on affine transformation", IEEE conf. International Congress on Image and Signal Processing, BioMedical Engineering and Informatics,2017.

[14] Lei Yu, "Large Screen Interactive Touch System Based on Affine Transformation", IEEE Conf. 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 544-548, 2017.

[15] RediaRedzuwan, "Affine versus projective transformation for SIFT and RANSAC image matching methods" , IEEE Conf. International Conference on Signal and Image Processing Applications (ICSIPA), pp. 447-451, 2015.

[16] Xuan Li, "Affine-Transformation Parameters Regression for Face Alignment", IEEE Signal Processing Letters, Vol. 23, Issue. 1, pp.55-59, 2018.

[17] Mankyu Sung,"Selective Anti-Aliasing for Virtual Reality Based on Saliency Map", IEEE Conf. International Symposium on Ubiquitous Virtual Reality, 2017, pp. 16-19.