IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

ANALYSIS OF FRACTAL IMAGE UNDER CAUCHY AND ITERATED SYSTEM

Meenakshi Choudhary[1], Prof. Sushma Lehri[2]

[1,2]Department of Computer Science, Institute of Basic Science Khandari, Agra (India)

ABSTRACT

A going round another picture makes picture of the footway of a purpose under semi-group of great changes. The idea as first started given by barnsley[1] has greatest, highest importance in image forced together, biological making copies to scale another areas of fractal geometry. In this paper, we put into use for first time higher iterations to work-place the part of having an effect equal to the input and nonlinear great changes on the range of experience of a purpose. different qualities of the worked out noted representatives have been had a discussion about to give an idea of the usefulness of work place in mathematical observations made different algorithms are given to work out the orbital picture and V-variable going round another picture. An algorithm to work out the distance between mages makes the work-place give motion. A short discussion about the fact in support of the Cauchy order of images is also given.

I. INTRODUCTION

The idea of computing is the going around another pictures the interesting associative property of semi-groups of affine great changes. These beautiful picture scan be constructed using the IFS (Iteratd FunctionSystem)foraffinegreatchanges.The new move forward of fractals, i.eV-variable fractals and super fractals has already taken their first steps. This new living-stage of fractals may be made using super IFS. For details one may have relation to important works needing payment to barnsley[35] and Kuijvenhoven [14]. V-variable fractals and super fractals are grousing increased techniques in able to use observations. Their branching out may be got clearly from the fact that they have possible unused quality applications in different areas of mathematical2 sciences and designing and making. Fractals have full of force applications in getting greater, stronger, more complete new technologies such as by numbers, electronic imaging to cover being full of living back knowledge, things not fixed Computing to part radio apparatuses at low band distance from side to side, reservoir 3 designing and making to design and saywhat will take place in the future producing from reservoirs 4.ElNaschie has stamped metal money fractals with physical design to be copied building in his nearby work [9,10]. in addition to designing and making applications5, the concept of V-variability in going round another picturesincreases the square measure of applications in knowledge processing machine giving clear, full picture, mathematics as well as biological making copies to scale. For a detailed knowledge of fractals, new living-stage of fractals and their properties, say something about to barnsley [1,3], devaney[6,7], Edgar [8], Hutchinson [12], Lapidusand Frankenhuysen [15], Mandelbrot [16,17] and Peitgen and Al6. [11,21]. We Begin with an example made up of a single great change to work out the range of experience of a picture, which gives us the visual idea of the range of experience of an image and the computational mathematics behind the image. Coming after, we move in the direction ofmorethanonegreatchangeandthe details of the knowledge processing machine putting into effect to work out the range of experienceofthepicture,whichmake surework-placeinterestingand simple, not hard to get through knowledge.While computing1 the range of experience of the picture, there may be some cases when the range of experience does not partly cover. We play or amusement that non-overlapping cases may be got changed into just touching or partly covering cases using higher iterations2 to give a true to likenessandnaturalcoming together pattern. In mathematical3 rules to make, a purpose is taken to be as a very solid (substance) group. In order to work-place the order of things produced by the algorithm4 described in the earlier new division of page, we need to work out the distance between things and for this purpose, we use the idea of Hausdorff distance between 2 very solid (substance) groups. In order that the order of things converges, we have to make certain whether the order of things is Cauchy. We briefly have a discussion an algorithm 4, which works out the Hausdorff distance between images to widen the use and applications in different fields of mathematics and knowledge processing machine giving clear, full picture. Apart from designing the Hausdorff distance, we cover the nearby idea of V-variability while computing 1 the orbital pictures. Some examples in company with algorithms5 are included to make clearby example or pictures the study. Experimental observations of thesegoinground another pictures has a discussion different aspects such as connectednesscriticalpoint, convergence of orbits and connection of first things. Our work-place is based upon the done again and again group event system (ifs) and is put into order into 2 parts trading with having an effect equal to the input and nonlinear 6 contractive done again and again systems.

II. SUPERIOR ORBITAL PICTURES

An orbital picture speaks to the photo of the circle of a picture. We can plainly comprehend circle of a picture by considering an IFS comprising of a solitary change f, say. Assume I(x,y) is the underlying picture (inset the photo of Fig. 1), where x and y are the pixel directions and f = (0.7x + 0.3, 0.7y + 0.3) is the change, which we have taken to figure the circle. In this,$S_{\{X;\, f\}}$ is the semi-group generated by f.. At that point, the circle of I is the succession of pictures acquired by the rehashed use of f, i.e., the grouping {I, f(I), f^2(I), . . .} (see Fig. 1).

This definition can additionally be stretched out to more than one change, which thus should create a semi-gathering. For case, the arrangement of contractive changes on X, where X is a metric space creates a semi-gathering.

We build the different orbital pictures and V-variable orbital pictures concerning predominant repeats. The technique o get the photos is sketched out by illustrations. In the first place, we consider an IFS semi-bunch $S_{(X;f1,f2)}$ produced by two changes f_1 and f_2. The components of the semi-gathering can take after an arrangement for their simple reference, i.e., the semi-amass produced by two changes f_1 and f_2 can be represented by the ordered set {f_1; f_2; f_{11}; f_{12}; f_{21}; f_{22}; . . .}. The issue emerges when the arrangement of pictures is covering or crossing. This issue can be overwhelmed by utilizing tops (union of pictures), characterized prior in the preliminaries, which causes us to pick the shade of those pixels for which more than one shading esteem is fulfilling the foundation (see Section 3.2). There might be some more cases like simply touching and picture tiling, and so on. A non-covering case can be changed over into simply touching or covering by diminishing the estimation of parameter s. The estimation of s at which the pictures will simply touch each other can be viewed as a basic point, i.e., the pictures will cover in the event that we diminish s further. The estimation of the basic point will likewise rely on the underlying picture, i.e., distinctive pictures will have diverse basic focuses for the same IFS. For illustration, the basic purpose of Fig. 2 will be at s = 0.9831, redress up to four spots of decimals, i.e., circles will cover for s < 0.9831.

We consider the accompanying arrangement of changes to create the orbital picture as for prevalent repeats. Figs. 4–6 portray the cases, where the underlying pictures are given in the relating inset. Consider the IFS semi-gather produced by f1 and f2, where changes are linear and contractive.

$$f_1 = \left(0.7x + 0.3, 0.7y + 0.3\right),$$

$$f_2 = \left(0.7x - 0.3, 0.7y + 0.3\right),$$

2.1. CAUCHY ARRANGEMENT OF PICTURES

Give {An} a chance to be the accumulation of pictures. Review that the succession {An} is Cauchy iff $h(A_n,A_m)\to 0$ at whatever point n, m→∞. We comment that the Hausdorff remove between any two pictures is assessed with the assistance of pixels of the comparing pictures. To be sure, the essential thought of assessing separations between two pictures originates from the work of [13,24,25] and others. Fig. 3 demonstrates that the succession created by orbital pictures utilizing the above arrangement of changes is a Cauchy grouping.

There are such a variety of calculations for demonstrating an arrangement to be Cauchy on PCs. For a point by point depiction, one may allude to [1,13,24,25]. In our paper, we build up a basic and hearty calculation to guarantee that the succession of pictures is Cauchy. Our calculation may not be extremely productive as far as time and space imperatives however it fills our need. We consider a portion of the orbital pictures and after an adequate number of cycles, we compute Hausdorff separations as for the Euclidean metric between any two of the items. We find that the arrangement is Cauchy and since in a total metric space, each Cauchy succession is merged, so our arrangement of pictures will positively subside into a stationary state. We build up a calculation (given toward the end) to figure different Hausdorff separations. For effortlessness, this calculation is appropriate as it were for dark picture on a white foundation. We ascertain the Hausdorff separate between some combine of pictures of orbital pictures (Fig. 3) utilizing the calculation. A gentle correlation is appeared in Table 1.

2.2. Superior orbital pictures

To begin the analysis we require an info/base picture I(x,y), where x and y are the pixel coordinates. We can standardize the pixel directions to make the picture fit on the PC screen. The accompanying strides delineate the calculation utilized as a part of the program to build the predominant orbital picture. Code of this calculation is given toward the end.

1. Consider the base image I(x,y) and apply the superior iteration on I, i.e., compute s f_1(I(x,y)) + (1-s)(x,y). Store this result in a temporary memory O_1, say.
2. Now apply s f_2(I(x,y)) + (1-s)(x,y) and store the result in another temporary memory O_2, say.
3. Take the union of I, O_1 and O_2 utilizing tops union and store this picture as yield picture O.
4. Clear the transitory pictures O_1 and O_2. Make yield picture O to function as new information picture for the following emphasis.
5. Rehash steps 1 through 4 for adequate number of times.

In spite of the fact that the orbital pictures produced by an IFS semi-amass are themselves assorted in nature yet their decent variety and utility could be expanded by presenting the idea of V-inconstancy and prevalent emphasess. We present here the unrivaled emphasis in the era of a V-variable orbital picture given by Barnsley [3]. There can be different approaches to build such pictures together with the advantages of mayhem amusement. We utilize the accompanying altered calculation to build the predominant V-variable orbital pictures. We consider the example of merging and give cases of direct and nonlinear changes.

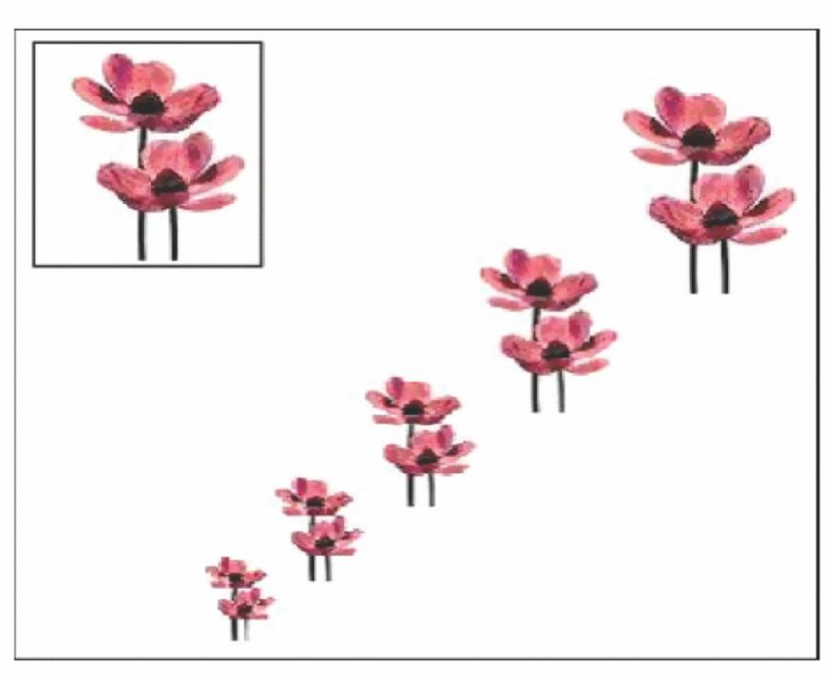

Fig.-1: The orbit of a picture

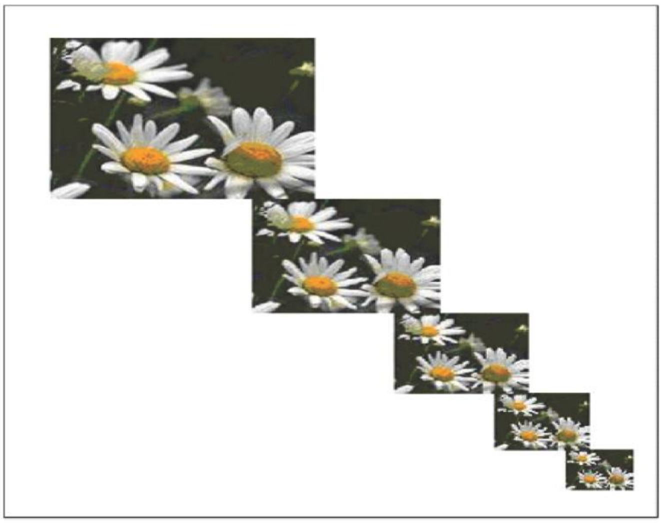

Fig.-2: Just touching superior orbit, critical point s=0.9831.

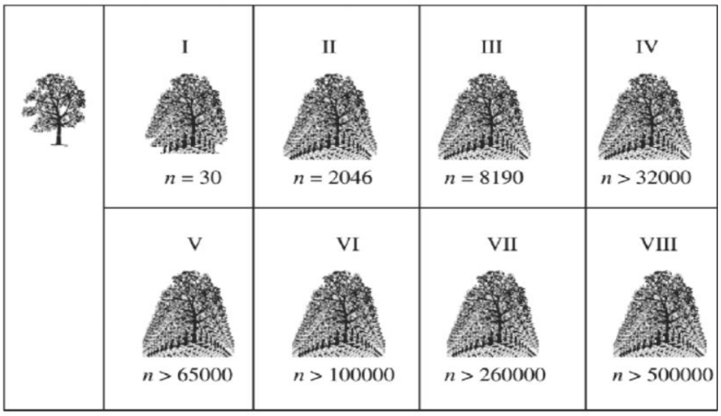

Fig.-3: A Cauchy sequence of orbital pictures with s=0.5 for different iterations (*n*).

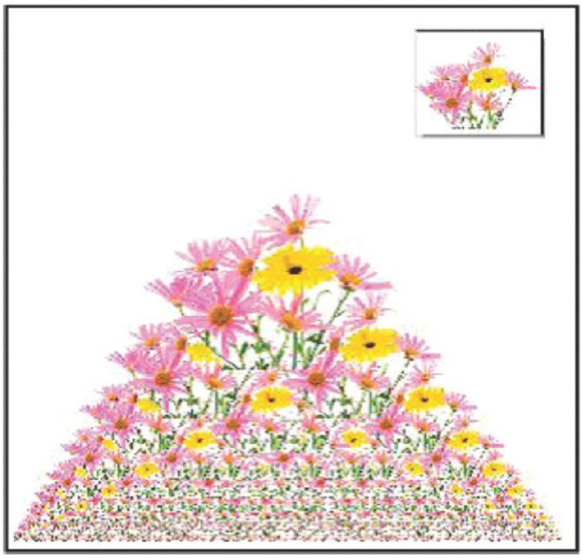

Fig.-4: Superior orbital picture for 10 iterations with s=0.9

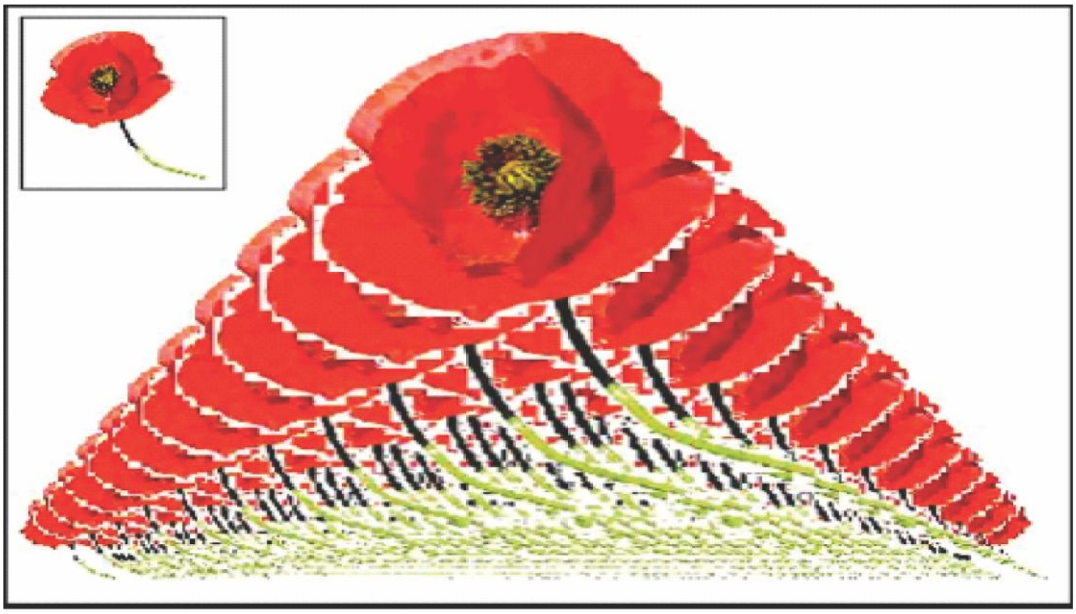

Fig.-5: Superior orbital picture for 10 iterations with s=0.5

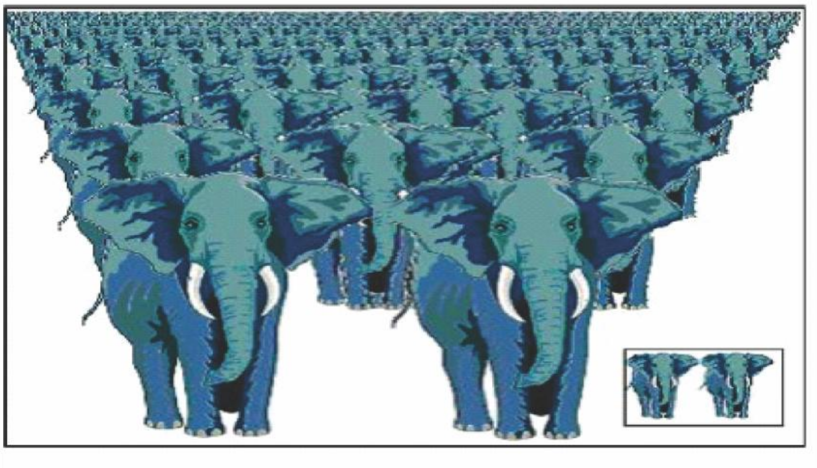

Fig.-6: Herd of elephants

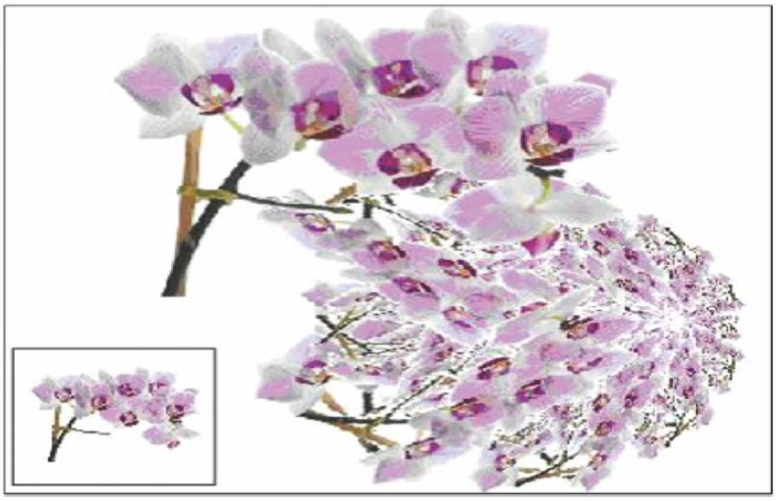

Fig.-7: One-variable superior orbital picture for 10 iterations with s=0.9

2.3. SUPERIOR 1-VARIABLE ORBITAL PICTURES

$$f_1^1(x,y)=\left(\frac{x}{2}-\frac{3y}{8}+\frac{5}{16},\frac{3y}{8}+\frac{3}{16}\right),\ p_1^1=\frac{1}{2},$$

$$f_2^1(x,y)=\left(\frac{x}{2}+\frac{3y}{8}+\frac{3}{16},-\frac{x}{2}+\frac{3y}{8}+\frac{11}{16}\right),\ p_2^1=\frac{1}{2},$$

$$f_1^2(x,y)=\left(\frac{x}{2}-\frac{3y}{8}+\frac{5}{16},-\frac{x}{2}-\frac{3y}{8}+\frac{13}{16}\right),\ p_1^2=\frac{1}{2},$$

$$f_2^2(x,y)=\left(\frac{x}{2}+\frac{3y}{8}+\frac{3}{16},\frac{x}{2}-\frac{3y}{8}+\frac{5}{16}\right),\ p_2^2=\frac{1}{2}.$$

Notice that, all the four transformations are linear and contractive. Details to obtain superior 1-variable orbital picturesare summarized in the following steps.

Give I a chance to be the info picture.

1.Pick one of the IFS F_1 or F_2 arbitrarily, say F_1.

2. Process and store both of these pictures as impermanent yield pictures O_1 and O_2 individually.

$sf_1^1\left(I\left(x,y\right)\right)+\left(1-s\right)\left(x,y\right)$ and $sf_1^2\left(I\left(x,y\right)\right)+\left(1-s\right)\left(x,y\right)$

3. Take tops union of I, O_1 and O_2 and store the subsequent picture as yield picture O.

4. Switch the information and yield pictures. Clear the yield picture. Make the picture O to fill in as new information picture I.

5. Repeat steps 1 through 4 for adequate number of times. We incorporate a few pictures produced by applying the above calculation (see Figs. 7–9). Beginning pictures are in the insets.

2.4. SUPERIOR 2-VARIABLE ORBITAL PICTURES

We stretch out our program further to get the pictures for 2-fluctuation. Consider the above-characterized super IFS {F_1,F_2}. Take any two information pictures, say I1 and I2. At each progression, we get two yield pictures. In our programming, we utilize the accompanying calculation.

1. Pick one of the IFS F_1 or F_2 haphazardly, say F_1 and one of the pictures, again picked arbitrarily, say I_1.

2. Register store both of these pictures as impermanent yield pictures O_1 and O_2 individually.

$sf_1^1\left(I\left(x,y\right)\right)+\left(1-s\right)\left(x,y\right)$ and $sf_1^2\left(I\left(x,y\right)\right)+\left(1-s\right)\left(x,y\right)$

1. Take tops union of I, O_1 and O_2 and store the subsequent picture as one of the final output picture FO_1.

4. Clear the transitory yield pictures O_1 and O_2.

5. Repeat steps 1 through 4 to compute the second last yield picture FO_2.

6. Now FO_1 and FO_2 will work as new input images. For the same, switch over the input and final output images and clear the output screens.

7. Repeat steps 1 through 6 for sufficient number of times.

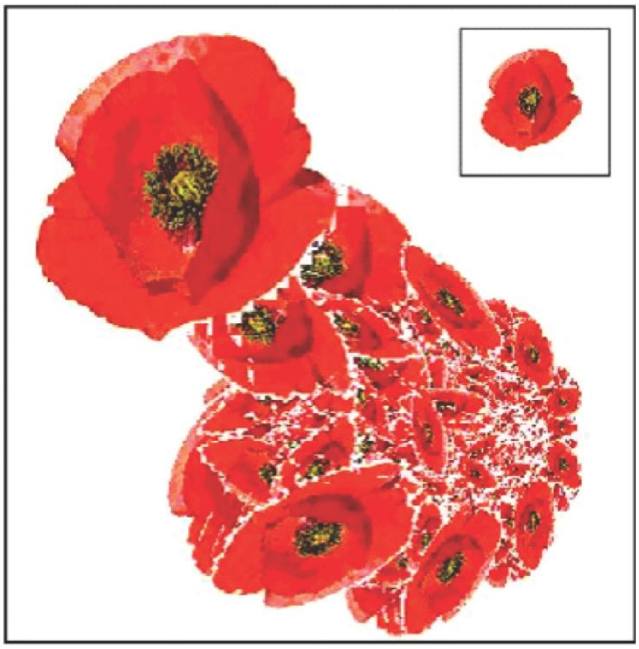

Fig.-8: One-variable superior orbital picture for 15 iterations with s=0.9

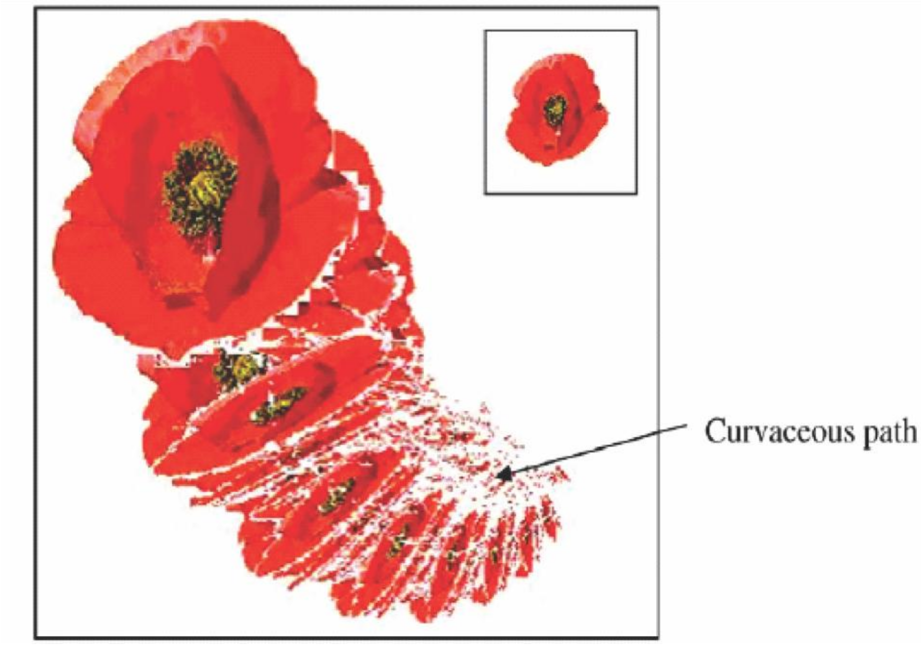

Fig.-9: One-variable superior orbital picture for 15 iterations with s=0.5

Figs. 10 and 11 are 2-variable superior orbital pictures generated by the above algorithm.

Following these steps, we obtained numerous orbital pictures of variability one and two for different values of s. For this purpose we have written a program in Visual C++ using Visual Studio 2005. We can modify the program to increase the variability.

A program can also be written in Matlab, OpenGL or any other standard language having the features of processingan image. Some of the selected images generated by this program are presented in this paper (see Figs. 1–13).

```
n*CChildView class is a CWnd derivative in Microsoft Visual C++ declared in childView.h. */
void CChildView::ImageConvergFun (CImage _ srcImg, CImage _ destImg, char funCase)
n*In this function, first argument refers to the input image. Second argument is the output image after applying
the function given in third argument. */
   {
    Double xScaleDown, yScaleDown, tempX, tempY, s = 0.1, ImageX = 0, ImageY = 0;
    intabsX, absY;
    byte r, g, b;
    COLORREF pixel;
    intMaxY = srcImg- >GetHeight (), MaxX = srcImg- >GetWidth ();
    for (int x = 0; x <MaxX;x++)
   {
    for (int y = 0; y <MaxY; y++)
        {
            pixel = srcImg- >GetPixel (x, y);
            r = GetRValue (pixel);
            g = GetGValue (pixel);
            b = GetBValue (pixel);
            if (pixel! = m_transparentColor)
            {
            xScaleDown = (x _ MaxX/2.0)/(MaxX/2.0);
            yScaleDown = (y _ MaxY/2.0)/(MaxY/2.0);
            switch (funCase)
            {
            case ’a’:
                tempX = 0.7 _ xScaleDown + 0.3;
                tempY = 0.7 _ yScaleDown + 0.3;
                ImageX = s _ tempX + (1 _ s) _ xScaleDown;
                ImageY = s _ tempY + (1 _ s) _ yScaleDown;
                break;
            case ’b’:
                tempX = 0.7 _ xScaleDown _0.3;
                tempY = 0.7 _ yScaleDown +0.3;
                ImageX = s _ tempX + (1 _ s) _ xScaleDown;
                ImageY = s _ tempY + (1 _ s) _ yScaleDown;
                break;
                default:
                    ;
                   }
                absX = (int)(ImageX _ (MaxX/2.0) + (MaxX/2.0));
                absY = (int)(ImageY _ (MaxY/2.0) + (MaxY/2.0));
                if (absX> 0 &&absX<MaxX&&absY> 0 &&absY<MaxY)
                    {
                        destImg- >SetPixelRGB (absX,absY, r, g, b);
                    }
                }
            }
        }
    }
void CChildView::OnToolsOptiimageconv ()
\* In this function, we have taken number of iterations, a blank image and an initial image form the user. At the
end, resultant image will be stored in place of blank image. */
   {
    UserInputDlgInputDlg;
    InputDlg.SetDlgMode (3);
    InputDlg.m_Iteration = 100;
    InputDlg.m_blankImage = _T (”C:ntempnnImageConvergnnBlank.bmp”);
    InputDlg.m_Image1 = _T (”C:nntempnnImageConvergnnInitImg.bmp”);
    if (InputDlg.DoModal ()! = IDOK)
    return;
    m_strImageName = InputDlg.m_blankImage;
    CStringstrImg1 = InputDlg.m_Image1;
    int iterations = InputDlg.m_Iteration;
    pImage- > Destroy ();
    pImage- > Load (m_strImageName);
    m_transparentColor = pImage- >GetPixel (0, 0);
    pImage- > Destroy ();
    pImage- > Load (strImg1);
    m_nImageSize = SIZE_ORIGINAL;
    OnToolsRefresh ();
    CWaitCursor wait;
    CImage _ptempImage1, _ptempImage2;
    ptempImage1 = new CImage ();
    ptempImage2 = new CImage ();
    for (int k = 0; k < iterations; k++)
    {
        ptempImage1- > Destroy ();
        ptempImage1- > Load (m_strImageName);
        ptempImage2- > Destroy ();
        ptempImage2- > Load (m_strImageName);
        ImageConvergFun (pImage,ptempImage1, ’a’);
        ImageConvergFun (pImage,ptempImage2, ’b’);
        MakeUnionOfImage (pImage,ptempImage1);
        OnToolsRefresh ();
        MakeUnionOfImage (pImage,ptempImage2);
        OnToolsRefresh ();
    }
     delete ptempImage1;
     delete ptempImage2;
      }
   void CChildView::MakeUnionOfImage (CImage _ unionImage,CImage _ imageToMerge)
   \*In this function, we are taking the tops union of two images passed in two arguments and the resultant
image is stored in the first argument. */
   {
    intMaxY = unionImage- >GetHeight (), MaxX = unionImage- >GetWidth ();
    byte r, g, b;
    COLORREF pixel;
    COLORREFunionBaseImage;
    for (int x = 0;x<MaxX;x++)
    {
       for (int y = 0;y<MaxY;y++)
    {
        unionBaseImage = unionImage- >GetPixel (x, y);
        if (unionBaseImage = m_transparentColor)
         {
        pixel = imageToMerge- >GetPixel (x, y);
        if (pixel! = m_transparentColor)
        {
            r = GetRValue (pixel);
            g = GetGValue (pixel);
            b = GetBValue (pixel);
            unionImage- >SetPixelRGB (x, y, r, g, b);
            }
        }
    }
   }
}
```

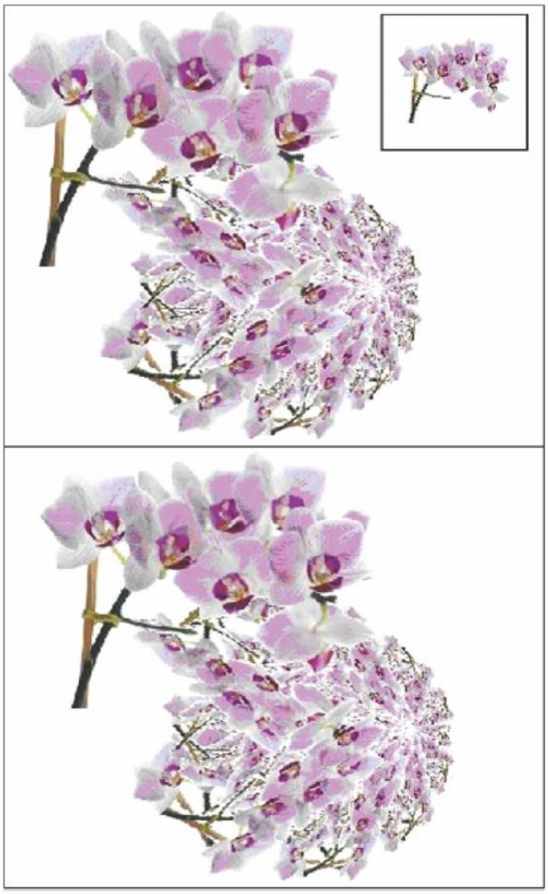

Fig.-10: Two-variable superior orbital picture for 15 iterations with s=0.9

2.5. NONLINEAR SUPERIOR ORBIT OF AN IMAGE

Orbital pictures can also be constructed when the transformations are nonlinear and contractive. For an illustration, we use the following set of nonlinear contractive transformations with equal probability:

$$G_1=\left\{\square;\, g_1^1, g_2^1\right\} \; and \; G_2=\left\{\square;\, g_1^2, g_2^2\right\},\; where \, \square := \left[0,1\right]\times\left[0,1\right]\subset R^2$$

$$g_1^1\left(x,y\right)=\left(\frac{x^2}{2},\frac{y^2}{2}\right),\; p_1^1=\frac{1}{2},$$

$$g_2^1\left(x,y\right)=\left(\frac{\left(x+1\right)^2}{4},\frac{\left(y-1\right)^2}{4}\right),\; p_2^1=\frac{1}{2},$$

$$g_1^2\left(x,y\right)=\left(\frac{\left(x-1\right)^2}{4},\frac{\left(y+1\right)^2}{4}\right),\; p_1^2=\frac{1}{2},$$

$$g_2^2\left(x,y\right)=\left(\frac{\left(x+1\right)^2}{4},\frac{\left(y+1\right)^2}{4}\right),\; p_2^2=\frac{1}{2}.$$

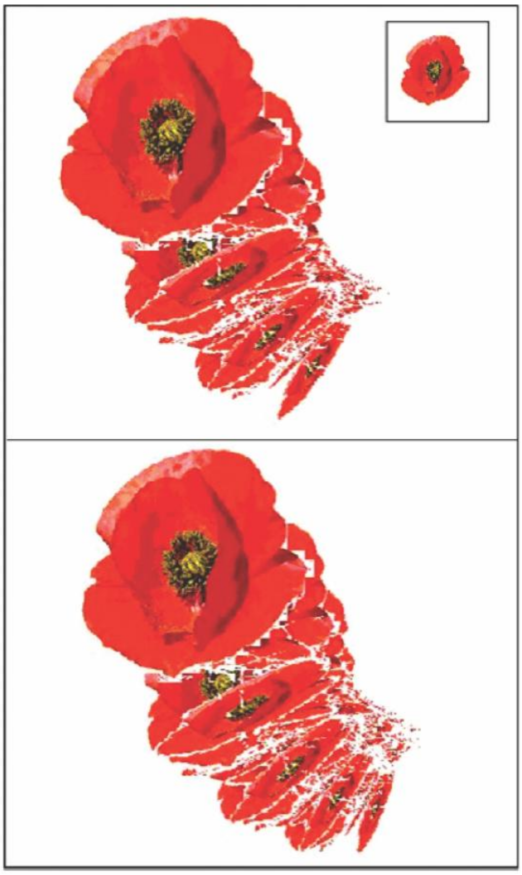

Fig.-11: Two-variable superior orbital picture for 15 iterations with s=0.5.

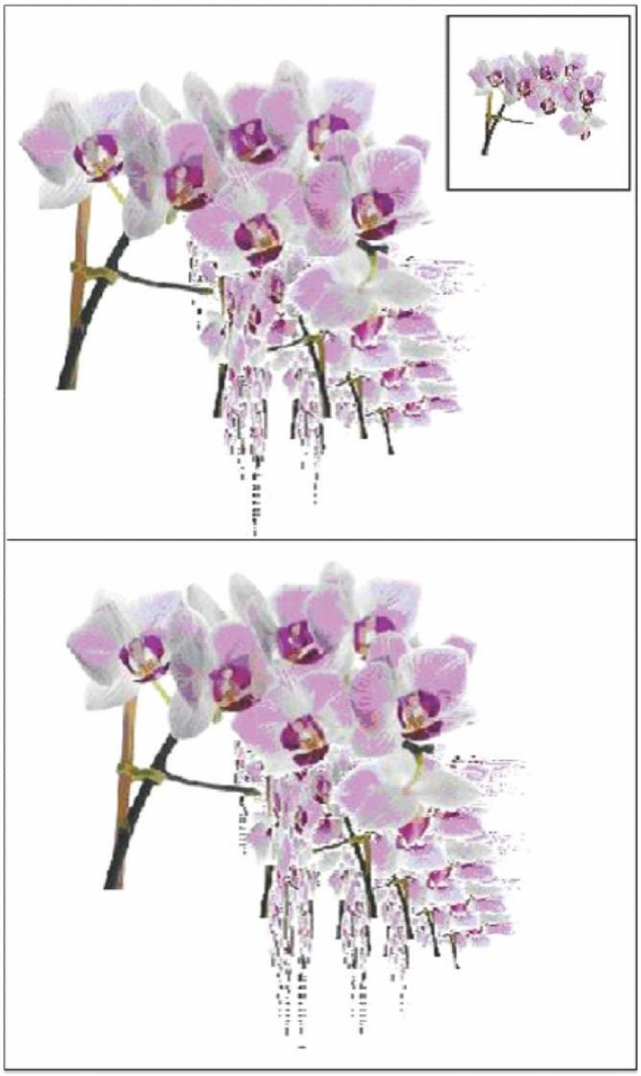

Fig.-12: Two-variable superior orbital picture for 15 iterations with s=0.5 for nonlinear transformations

III. CONCLUDING REMARKS

Orbital pictures have been created utilizing one-stage criticism process by and large called work emphases. Presenting a two stage criticism process, to be specific unrivaled iterative strategy, could additionally build the utility and territory of orbital pictures. We, in this paper, have created orbital pictures and V-variable orbital pictures for straight and nonlinear changes utilizing prevalent emphases. A large portion of these geometrical items seem to have comparable examples and normal attributes. It has been watched that the articles are delightful when the changes are direct, while models based on nonlinear changes in discrete progression have their own particular significance and utility. The parameter s in better cycles has an essential part than change over the non-covering design into simply touching or covering design. In Fig. 2, we have demonstrated that the question at s = 0.9831 is quite recently touching and will cover in the event that we diminish the parameters further. We have called this point as basic point as the question tends to change its character now. We have produced various figures utilizing our program. From those registered figures, we take those, which make them strike highlights. We have taken distinctive beginning articles to demonstrate the assorted variety of rising examples.

Further, while registering the orbital pictures, we acquire an attractor. It appears important to realize that the grouping of pictures is Cauchy or not. We have built up a calculation to gauge the separation between various match of pictures. Table 1demonstrates the calculation outline of separation between pictures. For instance, we discover the separation between picture An and B at serial number 2 in Table 1 is 71.45 pixels while the separation between picture An and B at serial number 4 is just about zero. Zeros in the remove section imply that the separation between relating pictures is near zero.

The rising example of pictures inspires us to examine about the joining. From Figs. 4 and 5, it appears that the objects merge to a straight line wherein the rising examples of Figs. 7 and 8 give a perspective of a vast umbrella like structure. We watch comparative joining designs for a similar arrangement of changes with s = 0.9 in Figs. 7 and 8, yet we too see that in Fig. 7, some white lines are showing up which are the limits of each blossom wherein no such white lines are obviously noticeable in Fig. 8. This demonstrates the decent variety of beginning pictures and we can comment here that the underlying pictures play an imperative part. We likewise watch a vast void area close to the joining zone where the question, which is following the orbital way, can't reach. In Fig. 9, example of the developing orbital pictures resembles a mellow bended track. Some scattered red pixels in the photo are following the same well proportioned way.

When we reach towards 2-inconstancy, the meeting idea of orbital pictures demonstrates a comparable example for the same estimation of s. This meeting example can be unique in the event that we change the estimation of s. Figs. 10 and 11 demonstrate this nature. We have included Figs. 12 and 13 to demonstrate the rising figures relating to nonlinear changes. We watch that these are intricate structures and forces more complexities. It could be fascinating to know whether they safeguard any numerical property while crossing through their circle.

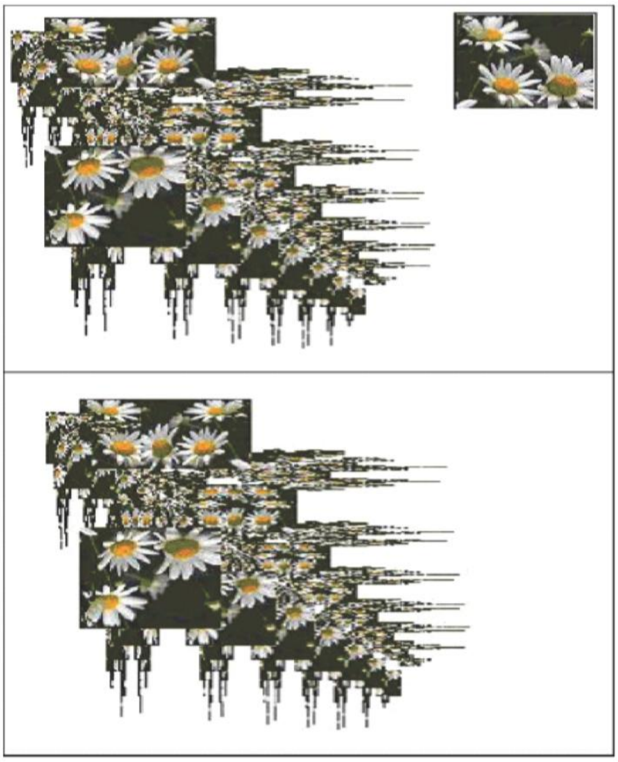

Fig.-12: Two-variable superior orbital picture for 15 iterations with s=0.9 for nonlinear transformations

Table-1

Hausdorff distances between images.

S.No.	Image A	Image B	Distance h (in pixels)
1	I	II	72.62
2	II	III	71.45
3	III	IV	0.0
4	III	V	0.0
5	III	VI	0.0
6	III	VIII	0.0
7	V	VII	0.0

REFERENCES

[1] Barnsley Michael F. Fractals everywhere. 2nd ed. Revised with the assistance of and with a foreword by Hawley Rising, III. Academic Press Professional:Boston, MA; 1993, MR1231795 (94h: 58101).

[2] Barnsley Michael F, Hutchinson John E, StenfloÖrjan. A fractal valued random iteration algorithm and fractal hierarchy. Fractals 2005;13(2):111–46.MR2151094 (2006b: 28014).

[3] Barnsley Michael F. Superfractals. Cambridge: Cambridge University Press; 2006. MR2254477 (2008c: 28006).

[4] Barnsley MF. Transformations between self-referential sets. Am Math Mon 2009;116(4):291–304.

[5] Barnsley MF, Hutchinson JE, StenfloÖrjan. V-variable fractals: fractals with partial self-similarity. Adv Math 2008;218(6):2051–88.

[6] Devaney Robert L. An introduction to chaotic dynamical systems. Menlo Park: The Benjamin/Cummings Publishing Co., Inc.; 1986. MR0811850 (87e:58142).

[7] Devaney Robert L. A first course in chaotic dynamical systems, theory and experiment. Advanced book program reading. MA: Addison-Wesley; 1992.MR1202237.

[8] Edgar Gerald A. Measure, topology and fractal geometry. New York: Springer-Verlag; 1990. MR1065392 (92a: 54001.

[9] El Naschie MS. Fractal black holes and information. Chaos Solitons Fractals 2006;29(1):23–35.

[10] El Naschie MS. Einstein’s dream and fractal geometry. Chaos Solitons Fractals 2005;24(1):1–5.

[11] EncarnacaoJL, Peitgen HO, Sakas G, Englert G. Fractal geometry and computer graphics. Berlin, Heidelberg: Springer-Verlag; 1992.

[12] Hutchinson John E. Fractals and self-similarity. Indiana Univ Math J 1981;30(5):713–47. MR0625600 (82h: 49026.

[13] Huttenlocher Daniel P, Klanderman Gregory A, Rucklidge William J. Comparing images using the Hausdorff distance. IEEE Trans Pattern Anal MachIntell 1993;15(9):850–63.

[14] Kuijvenhoven B. V-variable fractals: theory and implementation. <http://www.kuijvenhoven.net>.

[15] Lapidus Michel L. (Managing Editor), FrankenhuysenMachiel Van (editor). In: Proceedings of symposia in pure mathematics. 72, Parts 1 & 2: fractalgeometry and applications, a jubilee of benoit B. Providence: Mandelbrot, Amer Math Soc; 2004.

[16] Mandelbrot Benoit B. The fractal geometry of nature. San Francisco, Calif.: W.H. Freeman and Co.; 1982. MR0665254 (84h: 00021.

[17] Mandelbrot Benoit B. Fractals and chaos. The Mandelbrot set and beyond, Selecta Volume C. With a foreword by P.W. Jones and texts co-authored byC.J.G.Evertsz and M.C. Gutzwiller, Selected Works of Benoit B. New York: Mandelbrot, Springer-Verlag; 2004. MR2029233 (2005b:01054).

[18] Robert Mann W. Mean value methods in iteration. Proc Am Math Soc1953;4:506–10. MR0054846 (14,988f).

[19] Negi Ashish, Rani Mamta. A new approach to dynamic noise on superior Mandelbrot set. Chaos Solitons Fractals 2008;36(4):1089–96. MR2379372.

[20] Negi Ashish, Rani Mamta. Midgets of superior Mandelbrot set. Chaos Solitons Fractals 2008;36(2):237–45. MR2382154.

[21] Peitgen HO, Jürgens H, Saupe D. Chaos and fractals. New frontiers of science, 2nd ed. With a foreword by Mitchell J. Feigenbaum. New York: Springer-Verlag; 2004. MR2031217 (2004m:37061)).

[22] Rani M, Kumar V. Superior Julia set. J Korea Soc Math Edu 2004;8(4):261–77.

[23] Rani M, Kumar V. Superior Mandelbrot set. J Korea Soc Math Edu 2004;8(4):279–91.

[24] Shonkwiler R. An image algorithm for computing the Hausdorff distance efficiently in linear time. Inform Process Lett 1989;30(2):87–9. MR0983536(90c:68079.

[25] Shonkwiler R. Computing the Hausdorff set distance in linear time for any Lp point distance. Inform Process Lett 1991;38(4):201–7. MR1113350(92d:68115.

[26] Singh SL, Jain Sarika, Mishra SN. A new approach to superfractals. Chaos Solitons Fractals 2009;42:3110–20.