Generic Animation Method for Multi-Objects in IFS Fractal Form

Tedjo Darmanto[1,*], Iping Supriana[2] & Rinaldi Munir[2]

[1]Informatics Department, STMIK AMIK Bandung,
Jalan Jakarta 28, Bandung 40272, Indonesia
[2]Informatics Research Group, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132, Indonesia
*E-mail: tedjodarmanto@gmail.com

Abstract. Both non-metamorphic animation and metamorphic animation of objects or multi-objects in IFS fractal form as basic animation method can be implemented by a modified version of the random iteration algorithm as basic algorithm. In this way it is possible to create an attractive animation based on the combination of both animation methods as a generic animation method, as discussed in this paper. The proposed method can be used as an alternative for or as an extension of the existing animation methods in classical geometry, which are restricted to objects with regular forms. The proposed method can be used to produce more attractive media for interactively learning about the behavior of manipulated objects of any form.

Keywords: *animation method combination; fractal multi-object; generic method; IFS fractal model; metamorphic animation; non-metamorphic animation.*

1 Introduction

There are two kinds of animation methods for IFS fractal models. The non-metamorphic animation method is based on primitive operations to manipulate the animation object. There are two categories of manipulations that are suitable for animating objects naturally. The first one preserves the original form except the position, orientation and scale, and is called non-deformation manipulation. The second one is called deformation manipulation, because the form of the object is changed in a non-uniform way in one or both 2D directions. The other animation method is metamorphic animation, which also causes a deformation effect like the second category of the first method, but the manipulation operation is based on interpolation between the source and the destination of the objects as two different animation objects. Interestingly, this means that both deformation manipulations can be simulated by two different animation methods and the results are almost the same, which is discussed in this paper.

The main topic of this paper is how to integrate two different animation methods in a single mechanism or framework based on a combined method of

Received October 2nd, 2014, 1st Revision June, 24th, 2015, 2nd Revision November 2nd, 2015, Accepted for publication June, 14th, 2016.

non-metamorphic animation of fractal multi-objects and metamorphic animation of two fractal objects. For discussion of the animation model and the simulation, a windy-forest model was chosen. Two major events occur in the animation. The main event is swinging a collection of trees arbitrarily as in a forest. Some trees are swinging to the right while other trees in another part of the forest are swinging to the left over a certain distance and back in the opposite direction, so that both swinging events create an interesting weaving effect. At the same time, a cloud in the background is blowing from right to left over a certain distance and back again, thus going back and forth repeatedly.

This paper is divided into six sections. The first section is the introduction, followed by related works in Section 2. The third section discusses the generic animation algorithm of the IFS fractal model, followed by the model for the IFS fractal animation and a simulation of the IFS fractal animation. The last section is the conclusion.

2 Related Works

2.1 IFS Fractal

The IFS fractal model, one of the applicative models apart from the L-system fractal model [1] (not discussed in this paper), was introduced for the first time by Barnsley and Demko [2,3] based on Hutchinson's concept of 'self-similarity' [4]. By that time, fractals had become a new area of science, especially in mathematics and computer science. The term 'fractal' itself was coined for the first time by Mandelbrot [5], the father of the fractal model.

Basically, an IFS fractal model is based on a collection of affine functions that are self-contractive and therefore are called self-affine functions. A 2D self-affine function has six coefficients (**a, b, c, d, e,** and **f**) and can be expressed as the combination of a matrix and a vector in a mathematical expression [6,7] as described in Eq.(1):

$$w\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{1}$$

The coefficients in an IFS code set (**a, b, c, d, e** and **f**) as expressed in Eq.(1) are the affine transformation coefficients and can be classified into two kinds of transformations, linear and non-linear. Linear transformations are determined by the dimension factor of the object components relative to the dimension of the frame in the horizontal (**r**) and the vertical (**s**) direction, as expressed by Eq. (2). Non-linear transformations are determined by the orientation factor of the object components relative to the horizontal (**t**) and the vertical (**s**) direction, as expressed by Eq. (2) [7].

$$w\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}r*\cos t & -s*\sin u\\ r*\sin t & s*\cos u\end{bmatrix}*\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}e\\f\end{bmatrix} \tag{2}$$

2.2 Multi-objects

A fractal object, as the simplest form of any sort of object in an IFS fractal model, can be constructed by a random iteration algorithm based on IFS code as input, which is the representation of the affine transformation for each component inside the object. The simple fractal object can be used as a building block for a more complicated object, which is called a multi-object.

Basically, a fractal multi-object is a collection of fractal objects generated independently by means of a modified version of the random iteration algorithm. The algorithm is supported by random numbers that are partitioned, so the algorithm is called the partitioned-random iteration algorithm [8].

2.3 Morphing

The size of these objects cannot be measured by traditional methods based on Euclidean geometry, so it is measured by a method based on a fractal-based algorithm for metamorphic animation proposed by Chen, *et al.* [9]. Image morphing based on a pixel transformation approach as proposed by Rahman, *et al.* [10] descibes the transformation of pixels using their neighborhoods; this method is organized by the replacement of the pixel values of a source image and convolving the neighbor with the help of a mask that is fast and efficient for image morphing.

Zhuang, *et al.* [11] proposed a new morphing IFS fractal by calculating a local attractor's coarse convex-hull and selecting rotation matching between IFS's. The morphing procedure of two IFS fractal attractors is done by interpolating the parameters of the iterated function. Normally, a morphing animation deals with two different objects – the start and the target object – but if the morphing animation deals with more than two different objects, a new approach is needed, such as a method based on a family of multi-transitional IFS codes.

To generate a multi-object, a partitioned-random iteration algorithm can be used so the pixel population of an individual object as part of the multi-object can be controlled. A hybrid animation model based on a metamorphical interpolation model and a partitioned-random iteration algorithm has recently been proposed to simulate complicated multi-objects [12].

3 Generic Animation Method for IFS Fractal Models

The proposed generic animation method for IFS fractal models can be realized by integrating two kinds of animation algorithms. The first one is a modified version of the random iteration algorithm, i.e. the existing construction algorithm of the IFS fractal objects, as the core animation engine. The core engine is modified by means of a partitioned random generation mechanism, so the modified algorithm is called the partitioned-random iteration algorithm [12].

The second animation algorithm is an extended version of the first one with two core engines. The affine coefficients as the output of the two core engines are interpolated by means of an interpolation mechanism. The output of the second animation algorithm is one or many IFS code sets as the interpolation results between two different IFS code sets of the fractal object. Both animation algorithms use the same core animation engine, so the integration of both algorithms is feasible as the animation model's representation of the first, the second and the generic algorithm respectively, as can be seen in Figures 1-3.

Figure 1 illustrates the mechanism of animating a multi-object based on a scenario as input from a manipulator and an event-timing generator to drive the core engine. In the core engine, two parallel or independent processes occur. The first process deals with computing the pixel positions of the component objects based on the coefficients of the affine transform function. The second process deals with generating the random number ranges based on a probability factor. Both the coefficients and the probability factor are read from the IFS code. To generate more than one object as the components of a multi-object, the core engine is called iteratively based on the number of components of the multi-object. The scalar and the total number of random numbers determine the scale and density of the generated fractal objects.

Figure 2 illustrates the mechanism of animating two separate objects metamorphically based on an interpolator and an event-timing generator. The function of the interpolator is to interpolate two pixel positions coming from the two core engines. The two core engines here are the same as the core engine in Figure 1. The first core engine calculates the pixel positions and the random number range based on the source IFS code as the representation of the source object. The second core engine calculates the pixel positions and random number range based on the destination IFS code as the representation of the destination object. The same as for the multi-object animation engine in Figure 1, the scalar and the total number of random numbers are needed to determine the scale and density of the generated fractal objects.

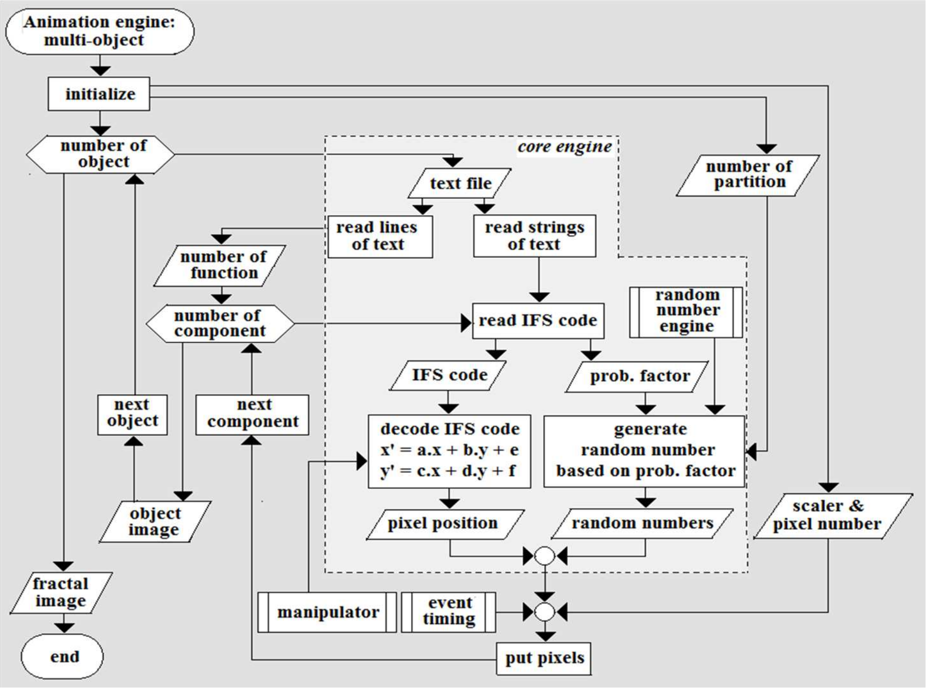


Figure 1 Flowchart of partitioned-random iteration algorithm as the animation engine for multi-objects in an IFS fractal model.

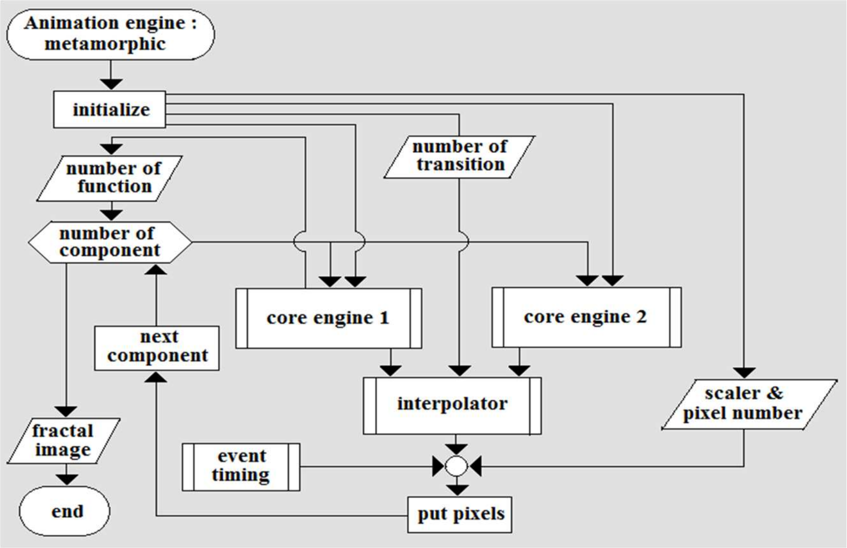


Figure 2 Flowchart of metamorphic animation engine for two IFS fractal objects.

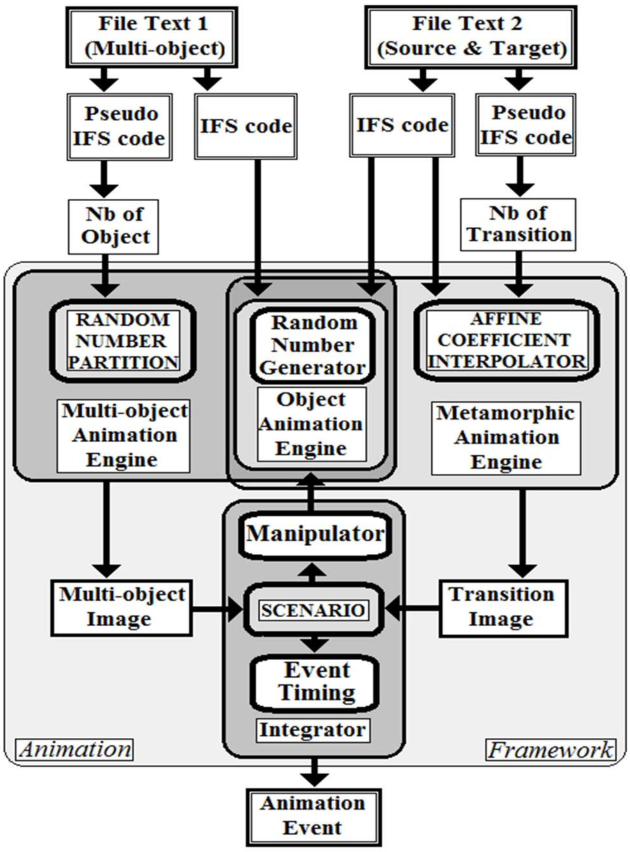


Figure 3 Block diagram of the animation framework as a representation of the generic animation algorithm in an IFS fractal model.

The IFS fractal animation framework as a representation of the generic animation algorithm was modeled as a block diagram to simplify the illustration instead of a flowchart diagram. Figure 3 illustrates the mechanism of animating a multi-object and two separate objects based on the same core engine as in the two previous animation engines, as shown in Figures 1 and 2 above. The framework consists of three parts: the multi-object animation engine, the metamorphic animation engine, and the integrator of both previous engines. The multi-object animation engine as the first part of the framework has the function to animate the multi-object as shown in Figure 1, while the metamorphic animation engine as the second part of the framework has the function to animate the two separate objects metamorphically, as shown in Figure 2. The third part of the framework is the integrator, which has a scenario mechanism based on two inputs coming from the first and the second part to drive the manipulator for the next event and the event timing to display the current event.

4 Model of IFS Fractal Animation

As mentioned in the introduction section, the animation model discussed in this paper represents a windy forest with two major events: a main event and a background event. The main event is the swinging-trees event and the background event is the blowing-cloud event. There are two scenarios for the windy-forest animation model. In the first scenario, the main event of the animation is modeled by the non-metamorphic animation method while the background event of the animation is modeled by the metamorphic animation method. In the second scenario, both events of the animation are modeled by the non-metamorphic animation method with almost the same result as from the first scenario, as already mentioned before.

To explain more clearly, there are two requirements that are different for both scenarios regarding the mechanism of animating the cloud. The first difference is in the number of IFS code sets of the cloud that are needed. In the first scenario, two IFS code sets of the cloud object need to be interpolated, while in the second scenario only one IFS code set of the cloud object needs to be interpolated. The second difference is in the need of a primitive shear operation in X direction to be executed or not. In the first scenario, the shear operation is needed but in the second one it is not as long as the series of shear operations has already been done when setting the destination IFS code for the cloud object.

The windy-forest animation model can be realized by a generic animation method represented by the IFS fractal animation framework based on the multi-object animation engine combined with the metamorphic animation engine, as explained in the previous section. The detailed mechanism and technique of this animation model are discussed in the 'Simulation' section. The reason why the windy-forest model was chosen is because this model can exhibit the superiority of fractal modeling over conventional animation modeling in generating complex animation objects and in the way of managing and controlling the manipulation operations, especially the deformation manipulation.

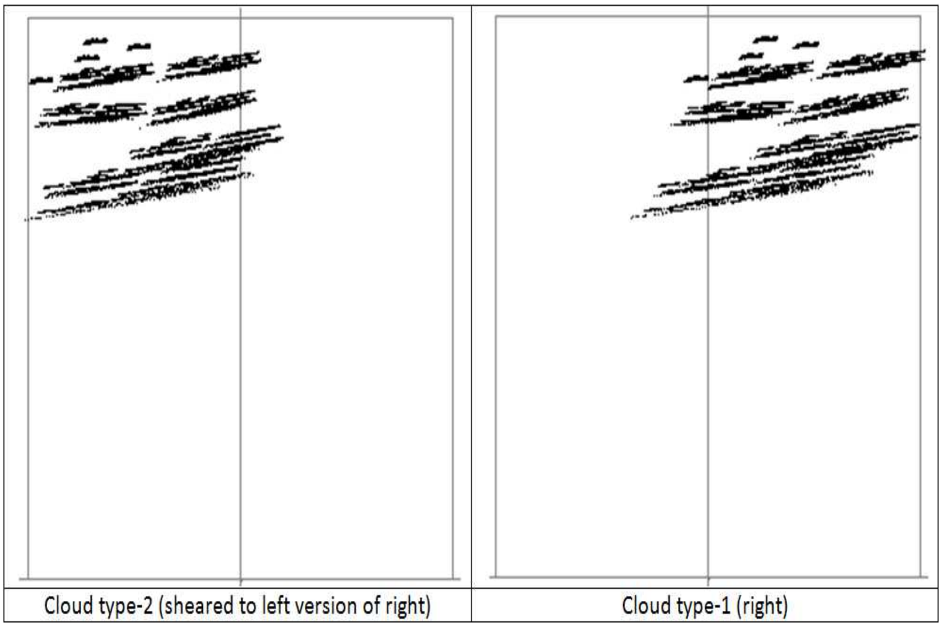
With a conventional method it is difficult to draw complex objects such as a tree and it is even more difficult to manage and control the small and gradual changes of the object by redrawing them in a slightly different position. On the other hand, in IFS fractal animation modeling, changing the affine coefficients of the object gradually before the represented object is constructed over and over can be done easily by repeatedly applying the shear operation with small fractions of the numerical parameters to the IFS code sets in certain periodical steps. As an example of an IFS code set, the IFS code of tree type-1 can be seen in Table 1.

Table 1 Typical representation of IFS Code (Tree-like Object).

a	b	c	d	e	f	Probability factor p
0.08	0.11	-0.02	-0.59	0.075	-0.60	0.24
0.308	0.312	-0.29	0.31	0.043	-0.26	0.18
-0.29	-0.30	-0.29	0.3	0.050	-0.34	0.18
0.764	0.197	-0.38	0.401	0.077	-0.48	0.24
0.201	0.25	-0.24	0.250	-0.02	-0.74	0.16

5 Simulation of IFS Fractal Animation

The cloud object (horizontally stretched version of the original) that is blowing from right to left, captured from the first image to the fifth image in Figure 4, shows a deformation effect that can be realized with a primitive shear operation of the affine function in the X direction to the left for the background event. The sheared effect can be observed in the areas marked by ovals and rounded rectangles. As long as the sheared effect occurs in one direction, this kind of effect can be realized also by a metamorphic animation method on two cloud objects as the source object (represented in the right image) and the destination object (the left image), or in a modified version (sheared to the left) as displayed in Figure 4 (left and right). The IFS code representations of both versions are displayed in Table 2 (left and right).



Cloud type-2 (sheared to left version of right)

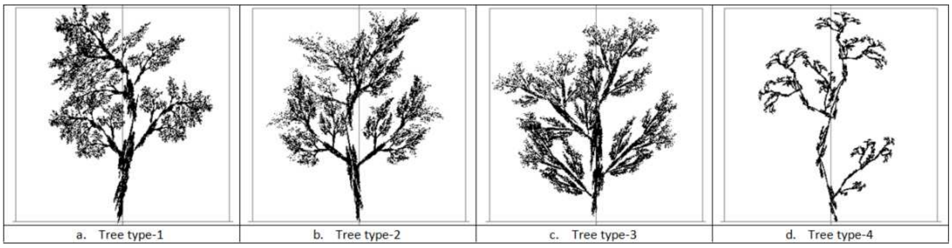
Cloud type-1 (right)

Figure 4 Two fractal cloud images. The left image is the sheared-to-left version of the right image in the X direction.

Table 2 Two IFS code sets of a Cloud in figure 4 (left and right).

a	b	c	d	e	f	a	b	c	d	e	f
0.87	-0.229	-0.08	0.2460	-0.237	-0.52	0.91	0.103	-0.08	0.206	0.023	-0.52
0.086	-0.027	-0.0	0.025	-0.474	-0.87	0.683	0.075	-0.07	0.154	0.222	-0.63
0.87	-0.229	-0.08	0.2460	-0.237	-0.52	0.459	0.05	-0.05	0.103	0.284	-0.75
0.086	-0.027	-0.0	0.025	-0.474	-0.87	0.44	0.039	-0.01	0.122	0.011	-0.73
0.87	-0.229	-0.08	0.2460	-0.237	-0.52	0.403	0.04	-0.03	0.101	0.345	-0.83
0.086	-0.027	-0.0	0.025	-0.474	-0.87	0.403	0.04	-0.03	0.101	0.081	-0.81
0.87	-0.229	-0.08	0.2460	-0.237	-0.52	0.086	0.003	-0.00	0.025	0.220	-0.93
0.086	-0.027	-0.0	0.025	-0.474	-0.87	0.086	0.003	-0.00	0.025	0.124	-0.94
0.87	-0.229	-0.08	0.2460	-0.237	-0.52	0.086	0.003	-0.00	0.025	0.089	-0.91
0.086	-0.027	-0.0	0.025	-0.474	-0.87	0.086	0.003	-0.00	0.025	-0.04	-0.87

A collection of trees in a forest as a multi-object can be generated by the partitioned-random iteration algorithm by means of the random partition mechanism for independently generating a tree as a component of a forest. For efficiency only four IFS code sets are used. The four components in fractal of a tree are represented by four IFS code sets, as shown in Figure 5. The first, the second and the third tree are duplicated by cloning techniques, so each tree becomes two trees in different sizes, horizontal orientations and positions, while the last tree is duplicated six times with the same technique, so the total number of trees is twelve. Each tree object is manipulated with a primitive shear operation of the affine function in the X direction, randomly applied either to the left or the right over a certain distance. The direction can be reversed, so in general the animation of a forest in a windy season can be realized with the trees swinging back and forth.



a. Tree type-1

b. Tree type-2

c. Tree type-3

d. Tree type-4

Figure 5 Four fractal images of four types of trees as the components of a windy forest fractal model.

The windy-forest animation model can be realized by IFS fractal modeling with at least two different models, as discussed in the previous section. Both models show the same general effect, so this simulation can be represented by either one of them. In this simulation, the animation is represented by five transition images as displayed in Figure 6. The animation exhibits a waving effect of trees swinging arbitrarily from the right to the left and vice versa as the main event and a non-linear deformation effect of a cloud moving from the right to the left as the background event, as can be seen in Figure 6 (marked by ovals, circles, rounded rectangles, rectangles and arrows).

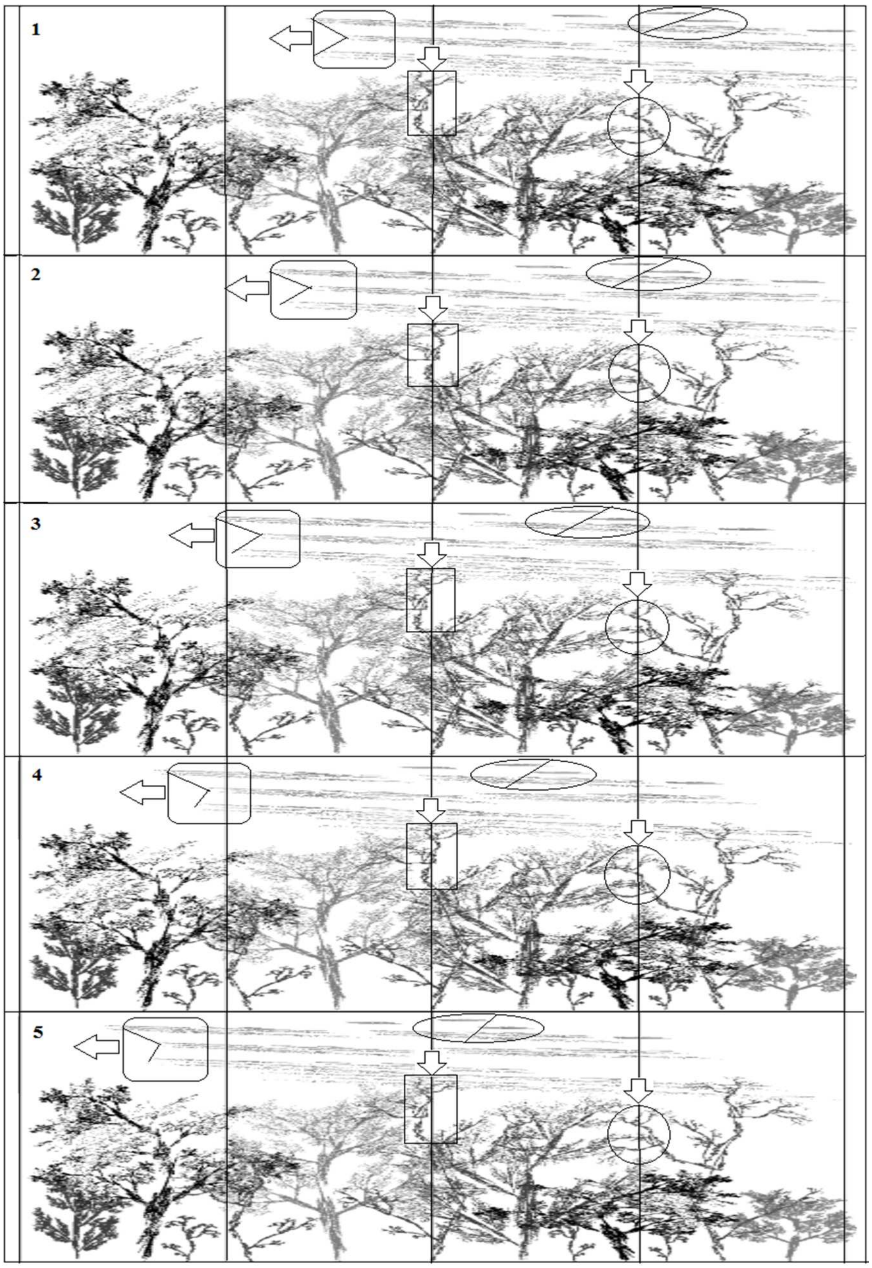


Figure 6 Sequence of transition images captured from windy-forest animation simulation.

6 Conclusion

Two different animation methods using IFS fractal modeling can be integrated and used to realize two events, the first event by non-metamorphic animation and the second event by metamorphic animation, as long as both methods have the same core animation engine inside.

The deformation effect of the non-metamorphic animation model in one direction can also be simulated by the metamorphic animation model using the extreme positions for the beginning and end versions as the source and destination of the animation objects.

Thus, a generic animation algorithm can be implemented to realize a combination of metamorphic and non-metamorphic animations of objects and multi-objects within a single animation framework.

References

[1] Lindenmayer, A., Fracchia, F.D., Hanan, J., Krithivasan, K. & Prusinkiewicz, P., *Lindenmayer Systems, Fractals, and Plants*, Springer Verlag, Berlin, 1992.

[2] Barnsley, M.F. & Demko, S., *Iterated Function Systems and the Global Construction of Fractals*, in Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, **399**(1817), pp. 243-275, 1985.

[3] Demko, S., *Construction of Fractal Objects with IFS*, Siggraph, **19**(3), ACM.0-89791-166-0/85/007/0271, pp. 271-278, 1985.

[4] Hutchinson, J.E., *Fractals Self-similarity*, Indiana University Mathematics Journal, **30**, pp. 713-747, 1981.

[5] Mandelbrot, B. B., *The Fractal Geometry of Nature*, W.H. Freeman and Company, San Francisco, 1982.

[6] Barnsley, M.F., *Fractals Everywhere 2nd Edition*, Morgan Kaufmann, Academic Press, San Diego, CA, 1993.

[7] Chang, H.T., *Arbitrary Transformation and Their Composition Effects for Two-Dimensional Fractal Sets*, National Yunlin University of Science and Technology, Taiwan, Image and Vision Computing, Publisher Elsevier, **22**(13), pp. 1117-1127, 2009.

[8] Darmanto, T., Suwardi, I.S. & Munir, R., *Hybrid Animation Model of Multi-object in Fractal Form based on Metamorphic Interpolation and Partitioned-Random Iteration Algorithms*, International Journal on Electrical Engineering and Informatics, **5**(3), pp. 285-296, 2013.

[9] Chen, C.B., Zheng, Y.P. & Sarem M., *A Fractal-based Algorithm for the Metamorphic Animation*, Animation, Information and Communication Technologies, ICTTA '06, **2**, pp. 2957-2962, 2006.

[10] Rahman, M.T., Al-Amin, Jobayer Bin Bakkre, M.A., Chowdhury, A.R. & Bhuiyan, M.A., *A Novel Approach of Image Morphing Based on Pixel Transformation*, 1-4244-1551-9/07 ©2007 IEEE, http://tajmilur-rahman.com/paper/Rahman2007ICCITBD.pdf (31 October 2014).

[11] Zhuang, Y., Xiong, Y. & Liu, F., *IFS Fractal Morphing based on Coarse Convex-Hull*, Information Technology and Artificial Intelligence Conference IEEE Joint International, pp. 225-228, 2011.

[12] Darmanto, T., Suwardi, I.S., & Munir, R., *Animation Simulation of Shooting Aircraft Events by Combination of Metamorphic Animation on Multi-object and on Object of IFS Fractal Model Based on Cloning and Collision Detection Techniques*, International Conference on Computer, Control, Informatics and Its Applications, 978-1-4799-4575-7/14 ©2014 IEEE, pp. 160-165, 2014.