A Methodology of Leaping Iteration for Drawing

Ming Jang Chen

Center for General Education, National Chiao Tung University, Taiwan; mjchen@mail.nctu.edu.tw

Abstract

A methodology of Leaping Iteration for drawing is introduced, followed by illustrations of using ovals to design elements of landscape painting, including views of mountain, cloud, tree, etc. The concept of "*Leaping Iteration*" is a variation of Leaping Iterated Function Systems (LIFS) that originated from "*self-similarity*" and "*self-replication*". This methodology provides an interface between mathematics and aesthetic. Moreover, it's consumption of computer resource is under control during the process of doing iteration.

Introduction

Iterated Function System (IFS), an interactive fractal generator, allows users to sketch first an outline of the desired fractal, and then cover it with deformed images of itself to achieve the collage, based on "Collage Theorem" proposed by Barnsley [1] et al. (1986), see also [3]. However, the number of intermediate objects grow exponentially fast while iterating, and this always exhausts computing resources before the necessary iterations are done. The concept of self-replication was used by Chen to design self-similarity [2] in *Bridges* 2014. The methodology of *Leaping Iterated Function System* (LIFS) first constructs a structure of the whole by parts (simple geometric shapes), converts it to an image, and then replaces each part with this image by means of *Structure Cloning Method* (SCM). The procedures will continue until the outcome is visually acceptable [2].

Algorithm LIFS (2014):

1. Let $W(x)$ be the function system represented by a generator.
2. Get $A_p = W(W^{p-1}(I))$, where I is an initiator (and p = 2 or 3).
3. Convert A_p into an image, and define a new generator $\overline{A_p}(x)$ by using the image of A_p as the only element in the pattern and the initiator I as the base.
4. Repeat Step 3, convert $\overline{A_p}(A_p)$ to an image, and define a new generator $\overline{A_{2p}}(x)$ by using this image as pattern and the initiator I as the base, then get the result $\overline{A_{2p}}(A_p)$.
5. Repeat the procedures

   $$\overline{A_p}(A_p), \overline{A_{2p}}(A_p), ..., \overline{A_{kp}}(A_p), ...$$

   Similar visual effects corresponding to $A_{2p}, A_{3p}, ..., A_{(k+1)p}, ...$ respectively.
   Stop iterating if outcome is visually satisfied.

Neither the outline of a given landscape element nor imagining the outcome from a simple generator is straight forward, and the outcome generated by the algorithm of LIFS totally depends on the generator itself. A new version of methodology is given below, which provides an interface between mathematics and aesthetics during iterations. This methodology can be executed on an add-in of PowerPoint. A landscape painting with mountain, cloud and grass, based on the methodology, is given in Figure 8.

A New Methodology for Drawing

Methodology of Leaping Iteration:

1. Construct a structure of the whole by parts.
2. Repeat the self-similar construction until the outcome is visually acceptable.
   2.1 Convert the current whole to an image.
   2.2 Construct the new whole by replacing each parts of the current whole with the image.
   A new whole is constructed and the structure remains unchanged.
   2.3 Tune the appearance of the whole, and the structure of the whole is also changed.

This methodology is illustrated in the following examples for drawing views of tree, and mountain respectively.

Example 1: Draw a tree. There are two generators for the construction of the structure of a tree. The first one is used to generate the tree structure in line segments by two direct iterations, see 3rd~5th figures in Figure 1. The second one is used to transform the tree in line segments to the tree in ovals, see 6th figure in Figure 1. The first step is to construct a structure of the tree by parts. In this example, we design a generator by a pattern and a base, see 1st figure in Figure 1, so that the pattern is composed of 4 line segments and a small branch. It is then followed by direct iterations by using 1st generator. Thus the result is considered as the structure of the tree in line segments, see 5th figure in Figure 1, which has 5^2 line segments in the structure.

Using line segments to sketch a structure is easy in general, but it is not easy to provide human interface for iteration and aesthetic purpose, the next step is to replace line segments by ovals by the 2nd generator. This means that there are 5^2 parts in the tree structure. Steps of showing how the algorithm grows in a self-similarity way by self-replication and how to tune the appearance of the tree can be found in 1st ~5th figures in Figure 2. In particular, 1st~3rd figures in Figure 2 show how Leaping Iteration generate the preliminary appearance in 5^2 parts. Based on the preliminary appearance, it is easier to tune some parts of the tree to adjust the appearance, see 4th and 5th figures in Figure 2. But Leaping Iteration is still required to get a new better looking, see 6th figure in Figure 2. Finally 7th and 8th figures in Figure 2 shows us the toning and composition effect.

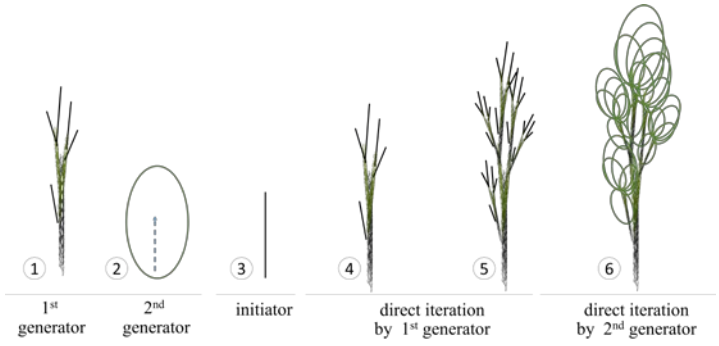

Figure 1: *Two generators for constructing structure of a tree*

Figure 2: *Grow self-similarity by self-replication and tune*

A mountain view can be drawn similarly by using Collage theory, we must outline the appearance of the mountain. However we need to answer a few problems, including how to describe the appearance of the mountain, what kind of geometric elements would be suitable to outline the mountain, how many components would be needed, as well as how to present the effect of light and shadow.

Figure 3: *These Chinese characters were written by famous calligraphers in history*.

The above Chinese characters [4] were written by famous calligraphers in history. We realize that the character "山" is a pictographic abstraction of mountain, and we propose to stroke "山" as the outline the mountain. What kind of geometric object is suitable? Oval may be a good choice because of no angular feature over it. If each stroke is represented by a geometric object, basically it needs four geometric objects, see Example 2. In order to present the effects of light and shadow, we fill the oval in transparent color and remove the frame of ovals, see Figure 4.

Example 2: To draw a mountain view. Motivated by the strokes of the Chinese character "山" (mountain), the generator of mountain view is designed, see 1st figure in Figure 4. After Iterating, the outcomes become more and more resemble to a mountain. As the ovals overlap, the transparency of the overlapped area is getting down while iterating, creating light effects from the outside to the center and forming various layers. In the Leaping Iteration, each view of mountain is composed of 4 components, there are affine transformation effects, though no affine transformation is executed.

Various generators for mountain views, together with their corresponding outcomes, can be found in Figure 5. Those outcomes are rich in variations, i.e., the role played by generators is very much sensitive. Each of these generators is composed of 4 ovals, these generators look similar but represent different function systems, even though all of them look like views of mountains. The first two look similar. The 3rd one is quite different in that its left side is cliff, right side looks like layers of mountain, while the 4th one also looks like building.

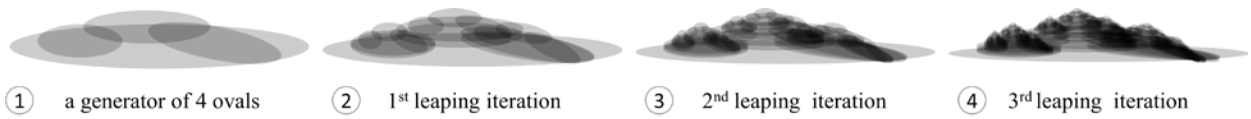

Figure 4: *A mountain view derived from 4 components.*

Figure 5: *Mountain views generated by similar generators*.

Example 3: Effects of light and shade, and layers of colors. In Figure 6, 2nd ~ 4th figure in Figure 6 represent only temporarily leaping iteration results by using the generator in 1st figure in four transparent ovals. The combination of the copies of the 3rd figure becomes a bigger mountain (see 5th figure in Figure 6). Through coloring a copy of the 5th figure in white and laying it on the mountain, they together form a new picture with cloud on the mountain, see the 6th figure. That is, the mountain and the cloud may have the same structure or they can be transformed into each other.

The picture shown in Figure 7 is composed of the intermediate outcomes of leaping iterative process. The fore ground of last picture in Figure 7 is from the 4th leaping, while the back ground is from the 2nd leaping. Four ovals can be used to design mountain, grass, cloud, etc. The composition of these elements enriches the connotation of landscape paintings, see Figures 6 and 7 respectively.

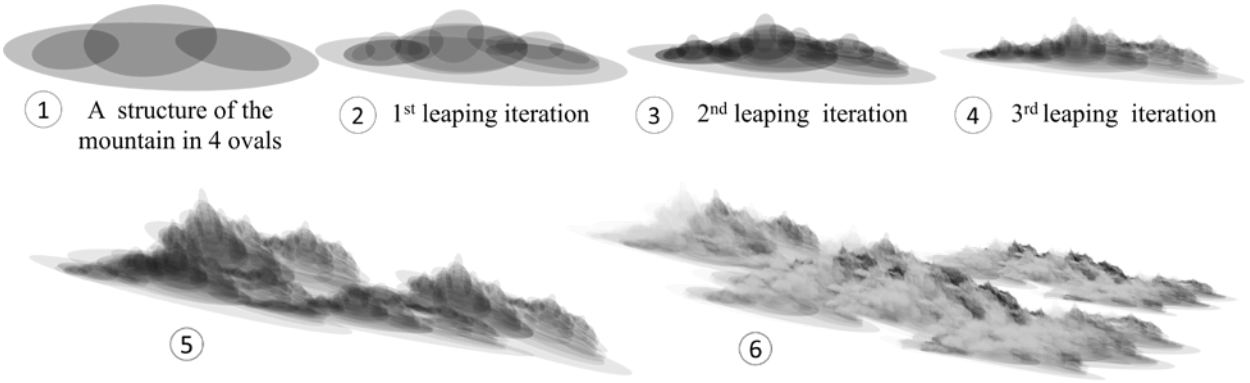

Figure 6: *Structure of the generator is composed of* 4 *ovals.*

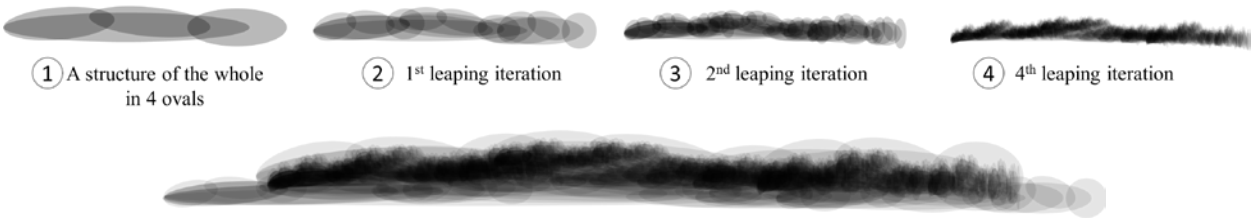

Figure 7: C*omposition of intermediate outcomes of leaping iterations.*

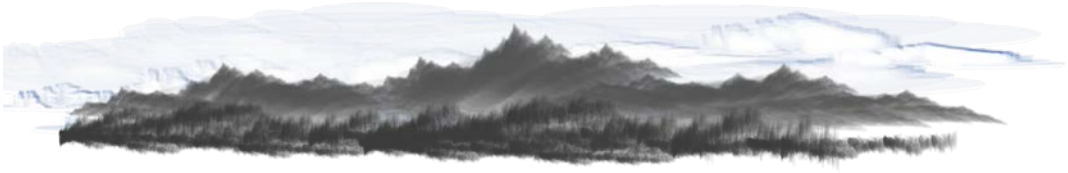

Figure 8: *A landscape painting with mountain, cloud and grass.*

Conclusion

Leaping Iteration not only inherits the characteristics of LIFS with computing consumptions under control, but also provides a manipulative environment for the interface between mathematics and aesthetics. We present in this paper the drawings of trees, a view of mountain, and cloud by Leaping Iteration starting from the arrangements of 4 ovals. There are still lots of new fractal patterns waiting for us to explore. Moreover, we show by example that light, shade, and layers of color effects can also be presented by Leaping Iteration. The intermediate outcomes of each iteration can also be used as elements for drawing purpose. We believe that the methodology of Leaping Iteration can make it possible to draw landscape painting in general on the screen.

References

[1] M. Barnsley. *Fractals Everywhere*. Academic Press, New York, 1989.

[2] M.Chen. "An Introduction to Leaping Iterated Function Systems." *Bridges Conference Proceedings*, Seoul, Korea, Aug. 14–19, 2014, pp. 345–348. http://archive.bridgesmathart.org/2014/bridges2014-345.html

[3] H. O. Peitgen, H. Jürgens, & D. Saupe. *Chaos and Fractals: New Frontiers of Science*. Springer-Verlag Inc., New York. 1992. Pp. 280.

[4] ISHUFA. http://ishufa.net/search/?q=凵.