Arbitrary Affine Transformation and Their Composition Effects for Two-Dimensional Fractal Sets

Hsuan T. Chang

Photonics and Information Laboratory
Department of Electrical Engineering
National Yunlin University of Science and Technology
Touliu Yunlin, 64045, Taiwan ROC
E-mail: htchang@pine.yuntech.edu.tw

Abstract

Fractal sets can be generated by the iterated function system (IFS) codes using the contractive affine transformation. This paper presents various geometric affine transformations and their composition effects for two-dimensional (2-D) fractal sets. Here the geometric transformations include translation, rotation, shearing, dilation/contraction, and reflection. First a hierarchical fixed point-searching algorithm is proposed to determine the original coordinates of a 2-D fractal set directly from its IFS code. Then the IFS code is modified according to the desired transformation. Instead of post processing the generated result, arbitrary affine transformation on the original fractal set can be directly obtained. On the other hand, the composite geometric transformations for 2-D fractal sets are also available. Finally, a complicated image frame can be synthesized by multiple 2-D fractal sets.

September 3, 2009

1 Introduction

Fractal techniques for computer graphics and image compression have been attracted a great deal of attentions more than two decades [1]. They can generate resolution-independent and self-similar patterns by the ways different from conventional techniques for computer graphics. On the other hand, the redundancy of self similarity in images is first utilized for image compression in fractal coding standard schemes. Their compression performance is comparable with other existing techniques. Therefore, the researches on fractal related techniques are very active because of the superiority in many applications.

Fractal techniques originate from that the natural images such as clouds, mountains, coastlines, plants, *etc.*, have very much self-similarity. That is, these natural images have the properties that their magnified subsets look like the whole set. Barnsley found that a finite set of specific contractive transform functions (CATs) of iterated function system (IFS) can generate a fractal image with the random iteration algorithm [1]–[3]. To find the CATs of a fractal object (i.e., to encode a fractal image), the fractal object is partitioned into several subsets, which differ to the entire object only in scale. Then the parameters denoting the CATs between the whole object and all subsets are calculated and recorded as the fractal code. The general form of a CAT W for a point located in the position (x, y) is [2]

$$W(\begin{bmatrix} x \\ y \end{bmatrix}) = \mathbf{A}\begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{b} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \tag{1}$$

where the coefficients $a, b, c, d, e,$ and f are real numbers and $|ad - bc| < 1$. An IFS can achieve a very high compression ratio since only few sets of parameters denoting the CATs are required. Its decoding algorithm is simple but both the encoding and decoding processes are computation-intensive in conventional methods [1].

Some methods [9], [13], [16] have been proposed to solve the inverse problem for encoding procedure and some parallel algorithms [5], [8], [14] are provided to speed up the decoding procedure. On the other hand, some optical architectures are proposed and implemented in [6], [7], [18] to show the parallelism and high-speed nature of optics. However, the methods above do not investigate the further transformation and the integration of the generated fractal images in the decoding stage. Although both issues can be performed *after* the fractal sets have been generated (i.e., use the post processing), both processes are troublesome and inefficient. Therefore, the IFS code should be modified such that the fractal sets can be directly generated with respect to the desired shapes and locations. On the other hand, the composition effects of different affine transformations should be explored. Furthermore, multiple fractal sets can be integrated into a single image frame. Two previous studies [4], [15] report the generation and manipulation of 2-D fractal sets. The former utilizes a multiple reduction copy machine and the latter introduces the non-linear IFS using the genetic programming.

Original IFSs can generate 2-D fractal sets. The scale and coordinates of a fractal set are unknown before the decoding stage is complete. However, its manipulation is important in the applications of computer graphics. To manipulate the 2-D fractal sets, one must first obtain the scale and coordinates information, then modify the original IFS code to perform arbitrary affine transformation of the original fractal sets. The manipulation of 2-D fractal sets is carried out using sequences of transformations. Translation, rotation, reflection, dilation/contraction, and shearing are defined by matrix. With matrix multiplications, their composition effects can be represented in a single matrix. An optical fractal synthesizer that can direct control the position, rotation, and scaling of fractal sets was recently proposed [17]. This method used an IFS mother function in the proposed optical fractal synthesize to manipulate the parameters for rotating, scaling, and positioning the generated patterns by optics. However, this OFS can decode only the fractal set that consists of a regular pattern or texture because the affine transformations in this architecture operate with equal probability. Also the image quality of the generated fractal sets is limited by the physical resolution of the optical system. In practical, the digital methods can generate more complicated fractal sets with the random iteration algorithm and easily perform the integration of multiple fractal sets.

Figure 1 shows the schematic diagram of the fractal encoding and decoding systems that we consider in this paper. For an original input image, the system goal is to obtain the output image that has been reformed according to the desired geometric transform. To achieve this goal, we propose a *hierarchical fixed-point searching* (HFPS) algorithm that can efficiently determine the original size and coordinates of a 2-D fractal set from its IFS code. This method is based on the Banach fixed-point theorem [1] and Pei's parallel algorithm [14]. First of all, we calculate the fixed points of all CATs and construct a convex contour that enclose all of the points and without overlap. The sub-level CATs of each CAT can be derived by the use of Pei's parallel algorithm. By hierarchically calculating the fixed points of sub-level CATs, the final contour is determined when no fixed points at the next-level CATs contribute the extension for the contour. Therefore, the original size of the decoded fractal image can be obtained. We next arbitrarily transform the original image by modifying its IFS code according to the original size and coordinates. Geometric affine transformations such as translation, dilation or contraction, rotation, shearing, and reflection for fractal sets can be obtained. The composition effects of different affine transformations are also investigated. Finally, we can synthesize a complicated image frame, which integrates multiple fractal sets generated with or without affine transformations.

2 HFPS Algorithm

The HFPS algorithm can automatically determine the original sizes and coordinates of fractal sets in 2-D space. With the determined information, the parameters in the IFS code can be modified so that the fractal set can be generated with arbitrary geometric transformation. Figure 2 shows the diagram of the proposed HFPS algorithm. The steps shown in this figure are described in the following subsections.

2.1 Fixed Points

First of all, the fixed point g_i for each CAT W_i is calculated. In fractal coding, a CAT W_i has a corresponding fixed point g_i, which can be calculated by [14]

$$g_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = (\mathbf{I} - \mathbf{A_i})^{-1}\mathbf{b_i}, \tag{2}$$

where $\mathbf{I}$ denotes the unit matrix. Fixed points are the important reference points in the encoding stage of fractal coding. For each subimage of a partitioned fractal image, the corresponding fixed point has to be selected in advance. To determine the parameters in the CAT of a subimage, the relationship between the fixed point and the corresponding point in the whole image are utilized [1]. At least three pairs of points in the subimage and the whole image are selected such that six parameters in the fractal code can be obtained by solving the six equations. It is easier to find the corresponding points if they are located at the image boundary. Therefore, there is a large probability that a fixed point and the corresponding point in the whole image are usually selected as located at the same point in the image boundary. That is, if we can determine the boundary points from the fixed points, then the size and the location of a 2-D fractal set can be obtained before the decoding process is complete. The HFPS algorithm is proposed based on this assumption.

2.2 Convex Contour Construction

The original size and the position of a 2-D fractal set shown in the 2-D space can be measure by the contour just covering the whole fractal set. Then the ranges along the x and y directions can be measured from the contour. To find the boundary points from the fixed points generated from different CATs, a convex contour constructed by connecting some of the fixed points is used. First of all, all the coordinates of the fixed points are examined to find the maximum and minimum x and y addresses. The fixed points located at the extreme addresses are connected to form a convex contour. Then other fixed points are checked whether or not they can make the contour be more convex. The fixed points that can make the contour more convex are left and their corresponding sub-level CATs are then determined.Otherwise, they are discarded and the corresponding CATs will not be used for determining their sub-level CATs.

2.3 Contour Extension Detection and Updating

Only the CATs whose fixed points contribute the extension of the convex contour are required to determine their sub-level CATs and the corresponding sub-level fixed points accordingly. From the final contour, we can determine the original size and coordinates of a fractal set directly from its IFS code. Let the level-one contour be the convex contour constructed from the fixed points of the original CAT.. For each CAT W_i, we then find its child (level-two) transformations $T_{ij}^{(2)}$ using Pei's parallel algorithm and the corresponding fixed points $g_{ij}^{(2)}$. Similarly, we plot the level-two contour based on the fixed points $g_{ij}^{(2)}$ and check whether or not the contour is extended. Once a fixed point $g_{ij}^{(2)}$ contributes the extension to the convex contour, we set it as the new contour point, which will be used to determine the CATs and the corresponding fixed points at the next level (level-three). The process above is recursively and hierarchically executed until the criterion that not any fixed point at the level $n+1$ contributes the extension to the contour at the level n. We thus obtain the size and the coordinate information of the fractal set from the information in the x and y ranges of the final convex contour. Four points located at the north, south, west, and east corners of the contour are taken to obtain their x and y coordinates. Therefore the size of the original fractal set can be determined by calculating the width at the horizontal direction, $x_{max} - x_{min}$, and the length at the vertical direction, $y_{max} - y_{min}$, respectively.

2.4 Summarization

The procedures of the proposed HFPS algorithm for the size determination of the fractal set are summarized as follows:

Step 1. The fixed points (x_i, y_i) for all CATs W_i are calculated [14]. Then we plot the level-one convex contour using the calculated fixed points.

Step 2. Apply Pei's parallel algorithm to generate $T_{ij}^{(2)}$ for each CAT W_i.

Step 3. For each $T_{ij}^{(2)}$, determine the corresponding fixed points $g_{ij}^{(2)}$. Compare with the fixed points in the previous level; check if the contour is extended or not.

Step 4. If the contour is extended, eliminate the fixed point at the previous level and refresh it by the new fixed points. Record the $T_{ij}^{(2)}$ such that we can employ Pei's parallel algorithm to generate the sublevel CAT $T_{ijk}^{(3)}$. Otherwise, stop extending the CAT at the next level, and prune this branch with index ijk.

Step 5. With the steps similar to steps 2∼4, recursively perform the propagation of $T_{ijkl\cdots}^{(n)}$ and the fixed-point calculation until that no extension of x and y ranges has occurred in the current step.

Step 6. The points in the final contour are used to obtain the maximum and the minimum values of the x and y coordinates, which specifies the original coordinates and size of the fractal set.

The change of coordinates for the CAT and the basic resizing and relocation effects on the fractal sets have been previously mentioned in Refs. [1] and [20], respectively. To perform arbitrary geometric transformation for original fractal sets, the parameters a, b, c, d, e and f in the IFS code will be modified based on the method shown in the next section.

3 Geometric Transformations

Let (x', y') denote the new coordinate system of the original point (x, y) in the 2-D fractal set. That is, $(x', y') = \Phi([x, y]^t)$ denotes the new coordinates of the point, where Φ denotes the coordinate transformation. Let $W'([x', y']^t)$ denote the same transformation as W but expressed in the new coordinate system. Then the relationship between the two CATs in two coordinate systems is expressed [1] by

$$W'(\begin{bmatrix} x' \\ y' \end{bmatrix}) = (\Phi \circ W \circ \Phi^{-1})(\begin{bmatrix} x' \\ y' \end{bmatrix}) = \mathbf{A}' \begin{bmatrix} x' \\ y' \end{bmatrix} + \mathbf{b}'. \qquad (3)$$

In the following subsections, we will investigate the modification of the IFS code in order to perform various geometric transformations. That is, the new matrices $\mathbf{A}'$ and $\mathbf{b}'$ will be determined according to the desired affine transformation.

3.1 Translation

Suppose that the original 2-D fractal set is shifted by h in x direction and v in y direction, respectively. In this case, $\Phi([x,y]^t)$=$([x+h, y+v]^t)$ and its inverse transform is $\Phi^{-1}([x',y']^t)$=$([x'-h, y'-v]^t)$. Therefore, the new CAT W' becomes

$$\begin{aligned} W'(\begin{bmatrix} x' \\ y' \end{bmatrix}) &= \mathbf{A}\begin{bmatrix} x'-h \\ y'-v \end{bmatrix} + \mathbf{b} + \begin{bmatrix} h \\ v \end{bmatrix} \\ &= \mathbf{A}\begin{bmatrix} x' \\ y' \end{bmatrix} + (\mathbf{I}-\mathbf{A})\begin{bmatrix} h \\ v \end{bmatrix} + \mathbf{b} \\ &= \mathbf{A}\begin{bmatrix} x' \\ y' \end{bmatrix} + \mathbf{b}', \end{aligned} \tag{4}$$

where $\mathbf{I}$ denotes the unit matrix. Apparently, the matrix $\mathbf{A}$ does not change and we can use the modified matrix $\mathbf{b}'$ to relocate the original fractal set, where

$$\mathbf{b}' = \begin{bmatrix} (1-a)h - bv + e \\ -ch + (1-d)v + f \end{bmatrix}. \tag{5}$$

That is, only the parameters e and f are modified for the translation on the original fractal set.

3.2 Dilation and Contraction

To change the scales of original fractal sets, the new CAT can be obtained by using Equation 3. Here three cases are discussed as follows:

1) Dilation or contraction in x-direction with the factor β_x, $\beta_x > 0$: The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} \beta_x & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a & b\beta_x \\ c/\beta_x & d \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e\beta_x \\ f \end{bmatrix}. \tag{6}$$

2) Dilation or contraction in y-direction with the factor β_y, $\beta_y > 0$: The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 1 & 0 \\ 0 & \beta_y \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a & b/\beta_y \\ c\beta_y & d \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e \\ f\beta_y \end{bmatrix}. \tag{7}$$

3) Simultaneous dilation or contraction in x-direction with the factor β_x and in y-direction with the factor β_y ($\beta_x \neq \beta_y$): The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} \beta_x & 0 \\ 0 & \beta_y \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a & b\frac{\beta_x}{\beta_y} \\ c\frac{\beta_y}{\beta_x} & d \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e\beta_x \\ f\beta_y \end{bmatrix}. \tag{8}$$

Note that when both factors are equal ($\beta_x = \beta_y = \beta$), Eq. (7) becomes

$$\Phi = \beta \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{A}' = \mathbf{A}, \mathbf{b}' = \beta \begin{bmatrix} e \\ f \end{bmatrix}.$$

3.3 Rotation

To rotate the original fractal set with a clockwise angle θ with respect to the origin and the original x axis. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

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

$$\mathbf{A}' = \begin{bmatrix} a\cos^2\theta - (b+c)\cos\theta\sin\theta + d\sin^2\theta & (a-d)\cos\theta\sin\theta + b\cos^2\theta - c\sin^2\theta \\ (a-d)\cos\theta\sin\theta - b\sin^2\theta + c\cos^2\theta & a\sin^2\theta + (b+c)\cos\theta\sin\theta + d\cos^2\theta \end{bmatrix}, \tag{9}$$

$$\mathbf{b}' = \begin{bmatrix} e\cos\theta - f\sin\theta \\ e\sin\theta + f\cos\theta \end{bmatrix}.$$

By using the modified IFS code, the generated fractal set can be rotated by any given angle θ. The fractal set can be arbitrary rotated based on any point after we shift the original origin to the assigned point.

3.4 Shearing

Three cases of shearing the original 2-D fractal set with respect to the origin are discussed as follows:

1) Shearing the fractal set in x-direction with the factor k_x. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 1 & k_x \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a + ck_x & b + (d-a)k_x - ck_x^2 \\ c & d - ck_x \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e + fk_x \\ f \end{bmatrix}. \tag{10}$$

2) Shearing in y-direction with the factor k_y. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 1 & 0 \\ k_y & 1 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a - bk_y & b \\ c + (a-d)k_y - bk_y^2 & d + bk_y \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e \\ ek_y + f \end{bmatrix}. \tag{11}$$

3) Shearing in x- and y-direction with the factors k_x and k_y, respectively. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 1 & k_x \\ k_y & 1 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} \frac{a-bk_y+ck_x-dk_xk_y}{1-k_xk_y} & \frac{(d-a)k_x+b-ck_x^2}{1-k_xk_y} \\ \frac{(a-d)k_y-bk_y^2+c}{1-k_xk_y} & \frac{-ak_xk_y+bk_y-ck_x+d}{1-k_xk_y} \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e + fk_x \\ ek_y + f \end{bmatrix}. \tag{12}$$

3.5 Reflection

The reflected fractal sets with four different directions are given as follows:

1) Reflection with respect to x axis. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a & -b \\ -c & d \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} e \\ -f \end{bmatrix}. \tag{13}$$

2) Reflection with respect to y axis. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} a & -b \\ -c & d \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} -e \\ f \end{bmatrix}. \tag{14}$$

3) Reflection with respect to $x = y$ axis. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} d & c \\ b & a \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} f \\ e \end{bmatrix}. \tag{15}$$

4) Reflection with respect to $x = -y$ axis. The transformation matrix Φ and the modified matrices $\mathbf{A}'$ and $\mathbf{b}'$ in the new CAT are:

$$\Phi = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{A}' = \begin{bmatrix} d & c \\ b & a \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} -f \\ -e \end{bmatrix}. \tag{16}$$

4 Composition Effects

Since all of the transformations shown above are linear, they can be arbitrarily merged to obtain their composition effects. A sequence of the transformations shown in above subsections can be performed by a single linear transformation defined by the product of the matrices. However, the translation and any sequence of transformations involving translation cannot be combined in this manner into a single matrix. If the *homogeneous coordinates* [19] are used to describe points in a plane, then the translation can also be accomplished through matrix multiplication, and any sequence of these transformations can be defined in terms of a single matrix. In homogeneous coordinates, a third component '1' is added to each coordinate, i.e., $[x, y, 1]^t$, and

$$W(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}) = \mathbf{M} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{17}$$

where $\mathbf{M}$ denotes the CAT represented in the homogeneous coordinates. The matrices corresponding to translation $\mathbf{L}$, rotation $\mathbf{R}$, dilation/contraction $\mathbf{D}$, shearing $\mathbf{S}$, and reflection $\mathbf{F}$ in homogeneous coordinates are given as follows:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \beta_x & 0 & 0 \\ 0 & \beta_y & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} 1 & k_x & 0 \\ k_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{F}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that the matrix $\mathbf{F}_x$ is corresponding to reflection with x axis. The matrices corresponding to other reflection directions can be obtained with similar methods. The new matrices corresponding to above transformations in new coordinates now become

$$\begin{aligned}
\mathbf{M_L} &= \mathbf{LML^{-1}} = \begin{bmatrix} a & b & (1-a)h - bv + e \\ c & d & -ch + (1-d)v + f \\ 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{M_R} &= \mathbf{RMR^{-1}} \\
&= \begin{bmatrix} a\cos^2\theta - (b+c)\cos\theta\sin\theta + d\sin^2\theta & (a-d)\cos\theta\sin\theta + b\cos^2\theta - c\sin^2\theta & e\cos\theta - f\sin\theta \\ (a-d)\cos\theta\sin\theta - b\sin^2\theta + c\cos^2\theta & a\sin^2\theta + (b+c)\cos\theta\sin\theta + d\cos^2\theta & e\sin\theta + f\cos\theta \\ 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{M_D} &= \mathbf{DMD^{-1}} = \begin{bmatrix} a & b\frac{\beta_x}{\beta_y} & e\beta_x \\ c\frac{\beta_y}{\beta_x} & d & f\beta_y \\ 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{M_S} &= \mathbf{SMS^{-1}} = \begin{bmatrix} \frac{a-bk_y+ck_x-dk_xk_y}{1-k_xk_y} & \frac{(d-a)k_x+b-ck_x^2}{1-k_xk_y} & e + fk_x \\ \frac{(a-d)k_y-bk_y^2+c}{1-k_xk_y} & \frac{-ak_xk_y+bk_y-ck_x+d}{1-k_xk_y} & ek_y + f \\ 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{M}_{\mathbf{F}_x} &= \mathbf{F}_x\mathbf{M}\mathbf{F}_x^{\mathbf{-1}} = \begin{bmatrix} a & -b & e \\ -c & d & -f \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned}$$

To combine different CAT matrices, we can directly multiply the corresponding matrices. For example, the dilation followed by a translation and then a rotation on a 2-D fractal set would be defined by $\mathbf{DRL}(\mathbf{x})$, where $\mathbf{x}$ denotes the new vector in the homogeneous coordinates shown above. The composite transformation $\mathbf{DRL}$ is desired to be described by a single matrix. Then the composite matrix $\mathbf{M_{comp}}$ can be determined in advance as follows:

$$\mathbf{M_{comp}} = (\mathbf{DRL})\mathbf{M}(\mathbf{DRL})^{-1}. \tag{18}$$

By using the parameters in this composite matrix, the fractal set is generated by reflecting the composition effects. In addition to the composition of multiple transformations for a 2-D fractal set, multiple fractal objects can be integrated in a single image frame. However, their original coordinates and scales may be different. To integrate different fractal sets into a single image frame, their original sizes and coordinates should be investigated by the proposed HFPS algorithm. Then their IFS codes can be modified in advance by the method shown in Section 3. That is, the IFS codes are modified according to the desired transformations so that the fractal sets can be generated with the desired transformations. With elaborate design, multiple fractal sets can be integrated to synthesize a complicated image frame.

5 Experimental Results

Figure 3 shows four original fractal sets generated from the IFS codes shown in Table 1: (a) Sierpinski triangle, (b) Fern, (c) Castle, and (d) Snowflake. First of all, the proposed HFPS algorithm is used to find the convex contours of four fractal sets. Figure 4(a)–(d) shows the determined convex contours, in which the fixed points for different CATs are represented by different symbols. The symbols '$\circ$,' '$\times$,' '$+$,' '$*$,' and '$\cdot$' denote the fixed points of the first, second, third, fourth, and fifth CATs, respectively, which are shown in Table 1. Note that the fixed points determined from the same or different CATs, or from different levels may overlap. The level-one contour is plotted with a dotted line and level-two one is plotted with a dashed line. As shown in Figure 4(a), (c), and (d), the fixed points of the level-two CATs cannot be used to extend the level-one contour. Therefore, only the fixed points of the level-one and level-two CATs are calculated. As shown in Figure 4(b), four level-two fixed points belonging to three level-two CATs are outside the level-one contour. Therefore, the level-three CATs of these level-two CATs and their 12 fixed points are further determined. Because the level-three fixed points cannot be used to extend the contour, the extension of the contour for Fern stops at level two. The reason is that, in the encoding stage, the fixed points are usually selected at the corner points of the image for easy encoding. Therefore, only the sub-CATs in very few levels are utilized in the proposed HFPS algorithm.

Table 2 shows that only few fixed points are calculated on the different levels such that we can determined the sizes and the locations for the four fractal sets. In conventional random iteration algorithm, thousands of iterations are required to generate a complete fractal set. If the same iteration numbers are used in the conventional random iteration algorithm, the size and location information cannot be obtained. Figure 5 shows the generated fractal sets in which the numbers of points shown in Table 2 are used in the random iteration algorithm. Obviously, they all are very different from those shown in Fig. 3. That is, we cannot obtain their correct sizes and locations from such few numbers of iterations in conventional random iteration algorithm. Therefore, the proposed HFPS algorithm is very efficient because of its low computation load. A simplified comparison between a special case of the proposed method and conventional post-processing scheme has been reported in Ref. [20]. The ranges of x and y addresses obtained from Figure 4 are the same as those shown in Table 3, which are obtained from the decoded images. In addition to the ranges of the fractal set, a coarse shape of the fractal set by connecting the fixed points in each level can also be obtained. Figure 6 shows the coarse shapes of the fractal sets in Figure 4. Therefore, one can roughly understand the original shapes of the fractal sets in advance. Previous work shows that the determination of fixed points are also useful for self-similarity detection [10] and patterns clustering [11], [12] in analyzing fractal patterns.

The experimental results for arbitrary affine transformation by using the methods shown in Section 3 are given as follows: First of all, a fractal set whose coordinates are arbitrarily shifted by (h, v) to a desired location is considered. The Fern generated from the modified IFS code is shown in Figure 7, which has been shifted by 2.18 in x direction so that all of the new coordinates are positive. The dilated and contracted Sierpinski triangles generated from the modified IFS codes are shown in Figure 8(a)–8(d). The corresponding dilation/contraction coefficients are (a) $\beta_x = 2$, $\beta_y = 1$, (b) $\beta_x = 1$, $\beta_y = 2$, (c) $\beta_x = \beta_y = 2$, and (d) $\beta_x = \beta_y = 0.5$, respectively. Figure 9(a)–9(d) shows the rotated Sierpinski triangles using the modified IFS codes. The corresponding rotation angles are (a) $\theta = 0^o$, (b) $\theta = 90^o$, (c) $\theta = 180^o$, and (d) $\theta = 270^o$, respectively. The sheared Sierpinski triangles generated from the modified IFS codesare shown in Figure 10(a)–10(d). The corresponding shearing factors are (a) $k_x = 0.5, k_y = 0$, (b) $k_x = -0.5, k_y = 0$, (c) $k_x = 0, k_y = 1$, and (d) $k_x = 0, k_y = -1$, respectively. Figure 11(a)–11(d) shows the Castle generated from the modified IFS code. They are the original Castle reflected with respect to different directions: (a) x axis, (b) y axis, (c)$x = y$ axis, and (d) $x = -y$ axis.

Figure 12 shows the original Fern and the Fern generated with the composite transformation in Equation 18. The coefficients in the composite transformation are $h = v = 10$, $\theta = 45°$, $\beta_x = \beta_y = 0.8$, and $k_x = k_y = 0.1$. Finally, Two image frames synthesized by multiple fractal sets are shown in Figure 13(a) and 13(b). Both synthesized image frames are composed of multiple fractal sets with elaborate design.

6 Conclusion

In conclusion, we proposed a hierarchical fixed point-searching algorithm that can determine the coarse shape, the original coordinates, and their scales of 2-D fractal sets directly from its IFS code. Then the IFS codes are modified to generate the new 2-D fractal sets that can be the arbitrary affine transformation of original fractal sets. The transformations for 2-D fractal sets include translation, scaling, shearing, dilation/contraction, rotation, and reflection. The composition effects of the transformations above can also be accomplished through the matrix multiplication and represented by a single matrix. Finally, different fractal sets with and without further transformations can be synthesized into a complicated image frame with elaborate design.

In our future work, we will extend the manipulation for 2-D fractal sets to three-dimensional cases. Three-dimensional fractal sets can provide more variety for the applications on computer graphics. Moreover, our work can be adapted to the object-based coding for grayscale images, whose concept has been used in MPEG-4 standard. The objects in image frames, coded by fractal techniques, can be transformed or manipulated upon our need with the method shown in this paper.

Acknowledgment

This research was partially supported by National Science Council, Taiwan, under contracts NSC 89-2213-E-324-050 and NSC 92-2213-E-224-047.

References

[1] M. Barnsley, *Fractal Everywhere, Second Edition*, Boston, MA: Academic Press, 1993

[2] M.F. Barnsley and A. Sloan, "A better way to compress image," *BYTE*, pp. 215–223, Jan. 1988

[3] M. Barnsley, and L. Hurd, *Fractal Image Compression*, Wellesely, MA: AK Peters, 1993

[4] A.K. Bisoi and J. Mishra, "Enhancing the beauty of fractals," *Proceedings of 1999 Third International Conference on Computational Intelligence and Multimedia Applications*, pp. 454–458, Setp. 1999

[5] H.T. Chang and C.J. Kuo, "A Fully parallel algorithm for fractal image decoding," *1997 Proceedings of International Conference on Circuits and Systems*, vol. 2, pp. 1277–1280, Hong Kong, June 1997

[6] H.T. Chang and C.J. Kuo, "An optical decoding architecture for the random iteration algorithm of iterated function system codes," *Optical Review,* vol. 1, no. 2, pp. 146–149, 1994

[7] H.T. Chang and C.J. Kuo, "Random iteration algorithm based optical parallel architecture for fractal image decoding by use of IFS codes," *Applied Optics,* vol. 37, no. 8, pp. 1310–1318, March 1998

[8] H.A. Cohen, "Deterministic scanning and hybrid algorithms for fast decoding of IFS encoded image sets," *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'92)*, vol. III, pp. 509–512, May 1992

[9] M. Kawamata, H. Kanbada, and T. Higuchi, “Determination of IFS codes using scale-space correlation functions,” *Proceedings of IEEE Workshop on Intelligent Signal Processing Communication Systems*, pp. 219–233, Taipei, Taiwan, Mar. 1992

[10] K. Kamejima, “Self-similarity detection in random texture via multi-scale image analysis,” *Proceedings of the 1997 36th SICE Annual Conference*, pp. 1273–1278, July 1997

[11] K. Kamejima, “Autonomous clustering of fractal patterns via Hausdorff potentials,” *Proceedings of the 1999 38th SICE Annual Conference*, pp. 1077–1082, July 1999

[12] K. Kamejima, “Nondeterministic kinetics associated with self-similarity processes with applications to autonomous fractal pattern clustering,” *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 6, pp. 890–895, Oct. 1999

[13] S. Pei, C. Tseng, and C. Lin, “Wavelet transform and scale space filtering of fractal images,” *IEEE Trans. on Image Processing*, vol. 4, no. 5, pp. 682–687, May 1995

[14] S. Pei, C. Tseng, and C. Lin, “A parallel decoding algorithm for IFS codes without transient behavior,” *IEEE Trans. on Image Processing*, vol. 5, no. 3, pp. 411-415, March 1996

[15] F. Raynal, E. Lutton, and P. Collet, “Manipulation of non-linear IFS attractors using genetic programming,” *Proceedings of the 1999 Congress on Evolutionary computation,* vol. 2, pp. 1171–1177, July 1999

[16] R. Rinaldo and A. Zakhor, “Inverse and approximation problem for two-dimensional fractal sets,” *IEEE Trans. on Image Processing*, vol. 3, no. 6, pp. 802–820, November 1994

[17] T. Sasaku, J. Tanida, and Y. Ichioka, “Direct control of fractal pattern generation on an optical fractal synthesizer,” *Applied Optics*, vol. 39, no. 17, pp. 2959–2964, June 2000

[18] J. Tanida, A. Uemoto, and Y. Ichioka, “Optical fractal synthesizer: concept and experimental verification” *Applied Optics*, vol. 32, no. 5, pp.653–658, February 1993

[19] G. Williams, *Linear Algebra with applications*, third edition, Chapter 7, pp. 342–344, Wm. C. Brown Publishers, Dubuque, 1996

[20] Hsuan T. Chang, “Resizing, relocation, and quantization effects on fractal images,” *Optical Engineering* (SCI), vol. 40, no. 6, pp. 941–951, June 2001

Table 1: IFS codes for Sierpinski triangle, Fern, Castle, and Snowflake.

Image	W	a	b	c	d	e	f	p
Sierpinski triangle	1	0.5	0	0	0.5	0	0	0.33
	2	0.5	0	0	0.5	1	0	0.33
	3	0.5	0	0	0.5	0.5	0.5	0.34
Fern	1	0	0	0	0.16	0	0	0.01
	2	0.2	-0.26	0.23	0.22	0	1.6	0.07
	3	-0.15	0.28	0.26	0.24	0	0.44	0.07
	4	0.85	0.04	-0.04	0.85	0	1.6	0.85
Castle	1	0.5	0	0	0.5	0	0	0.25
	2	0.5	0	0	0.5	2	0	0.25
	3	0.4	0	0	0.4	0	1	0.25
	4	0.5	0	0	0.5	2	1	0.25
Snowflake	1	0.333	0	0	0.333	0.333	0	0.2
	2	0.333	0	0	0.333	0	0.333	0.2
	3	0.333	0	0	0.333	0.333	0.333	0.2
	4	0.333	0	0	0.333	0.333	0.666	0.2
	5	0.333	0	0	0.333	0.666	0.333	0.2

Table 2: The number of the determined fixed points for four fractal sets.

Image	Level 1	Level 2	Level 3	Total number
Sierpinski triangle	3	9	0	12
Fern	4	16	12	32
Castle	4	26	0	20
Snowflake	5	25	0	30

Table 3: The measured x and y ranges and the size of four fractal sets.

Image	$x_{\min}$	$x_{\max}$	$y_{\min}$	$y_{\max}$	size
Sierpinski triangle	0	2	0	1	2×1
Fern	-2.18	2.66	0	10.0	4.8×10
Castle	0	4	0	2	4×2
Snowflake	0	1	0	1	1×1

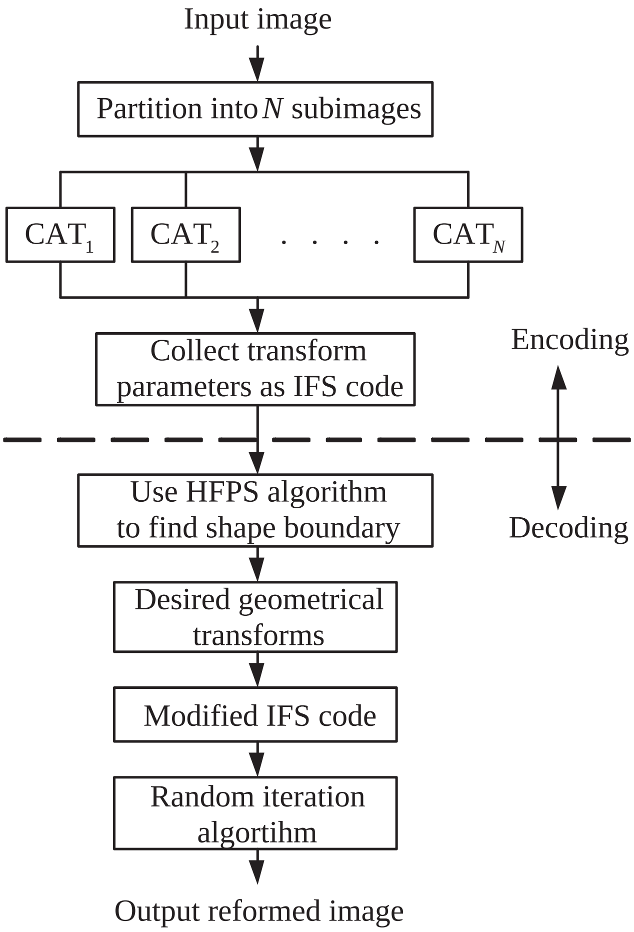

Input image
Partition into N subimages
CAT1
CAT2
. . . .
CATN
Collect transform parameters as IFS code
Encoding
Decoding
Use HFPS algorithm to find shape boundary
Desired geometrical transforms
Modified IFS code
Random iteration algortihm
Output reformed image

Figure 1: The schematic diagram of the encoding and decoding systems

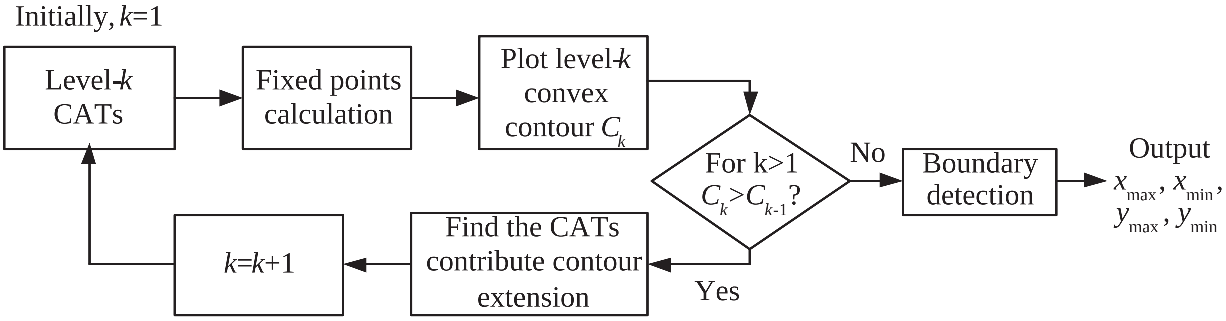

Initially, k=1
Level-k CATs
Fixed points calculation
Plot level-k convex contour Ck
For k>1 Ck>Ck-1?
No
Yes
Boundary detection
Output xmax, xmin, ymax, ymin
Find the CATs contribute contour extension
k=k+1

Figure 2: Block diagram of the proposed HFPS algorithm.

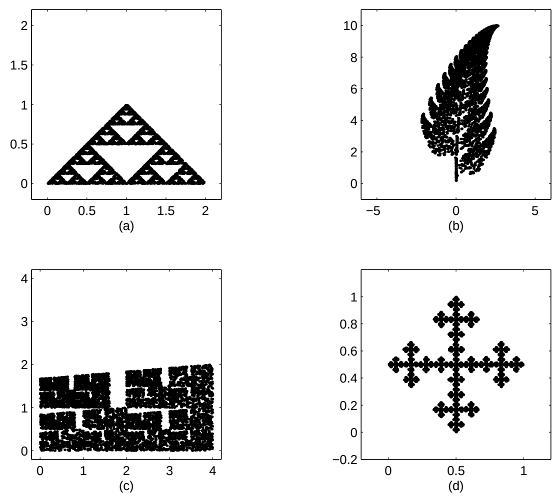
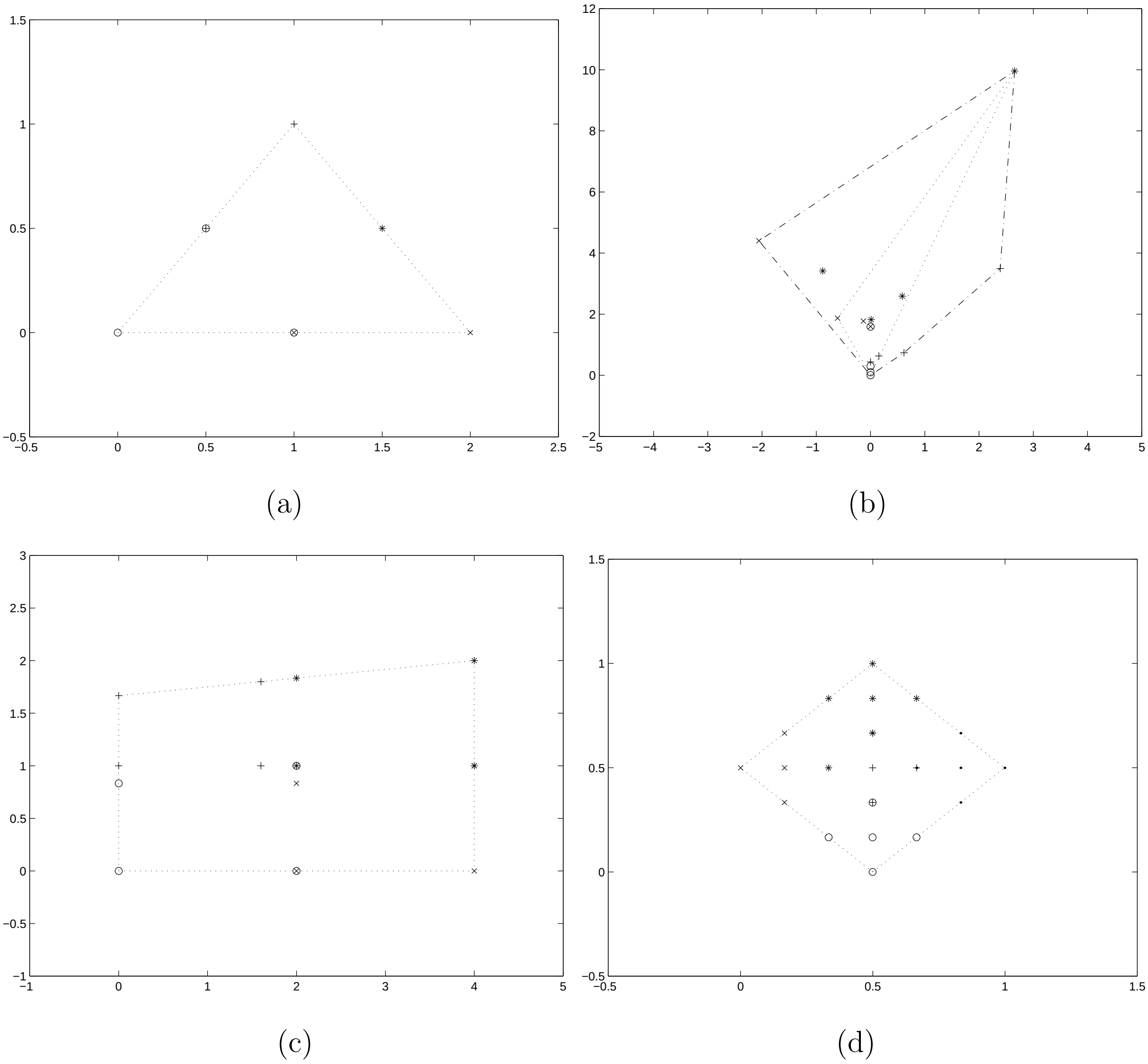


Figure 3: The decoded fractal sets by using the random iteration algorithm: (a)Sierpinski triangle, (b) Fern, (c) tree, and (d) Snowflake.



(a)

(b)

(c)

(d)

Figure 4: The determined contours based on the proposed HFPS algorithm: (a) Sierpinski triangle, (b) Fern, (c) Castle, and (d) Snowflake.

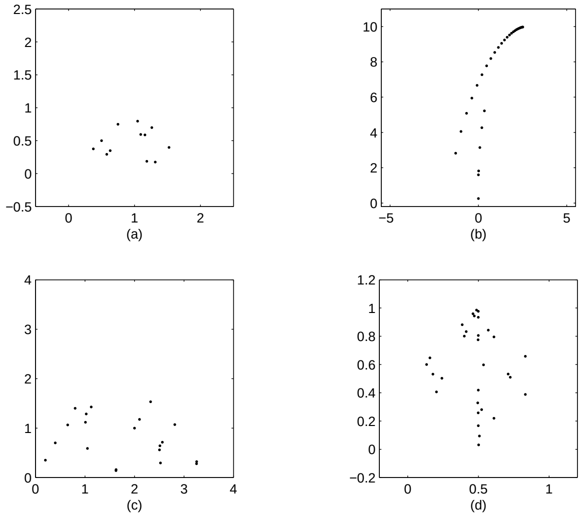


Figure 5: The fractal sets generated by the random iteration algorithm only using the numbers shown in Table 2.

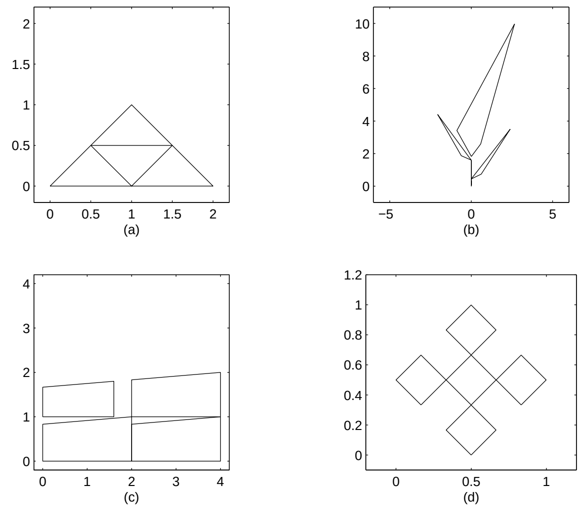


Figure 6: The coarse shapes of the fractal sets in Figure 3.

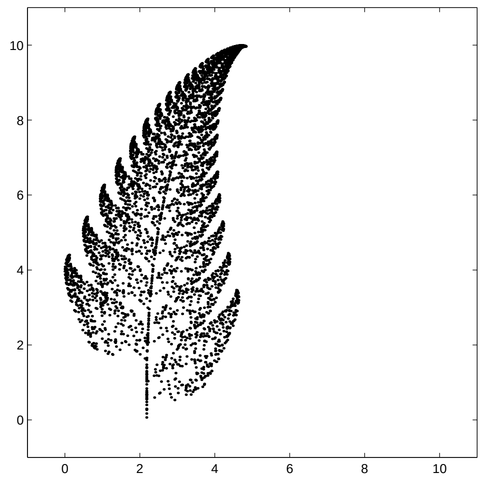


Figure 7: The shifted Fern generated by the modified IFS code.

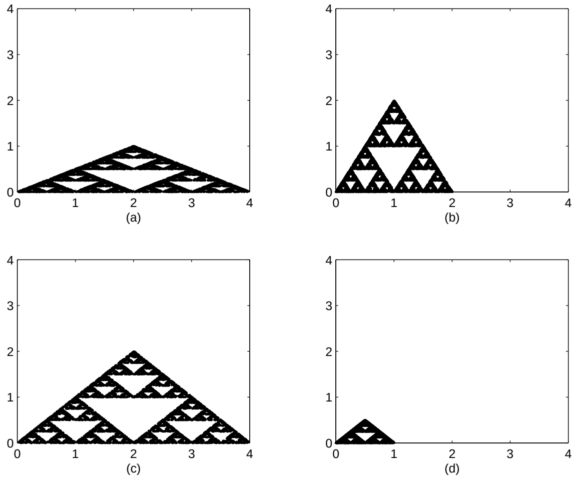


Figure 8: The dilated and contracted Sierpinski triangles generated by the modified IFS codes.

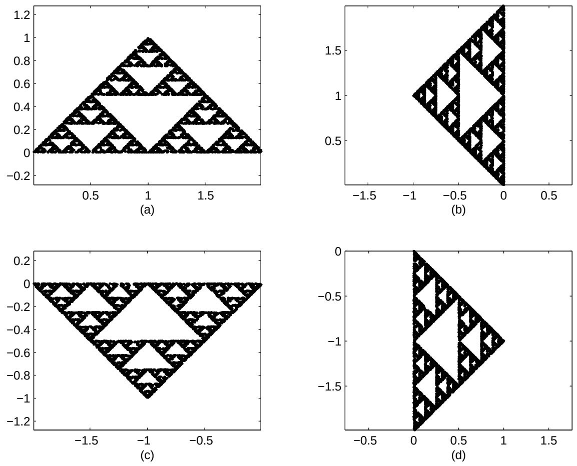


Figure 9: The rotated Sierpinski triangles generated by the modified IFS codes.

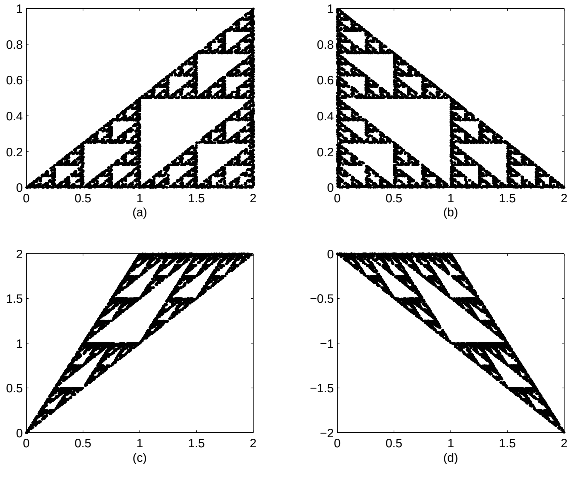


Figure 10: The sheared Sierpinski triangles generated by the modified IFS codes.

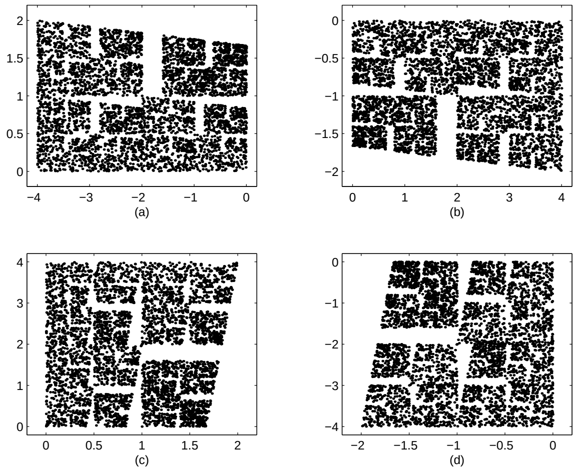


Figure 11: The Castle with different reflection directions, which are generated by the modified IFS codes .

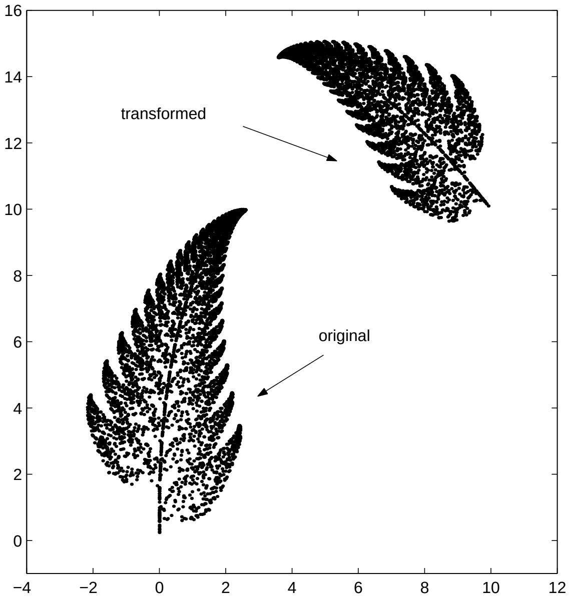
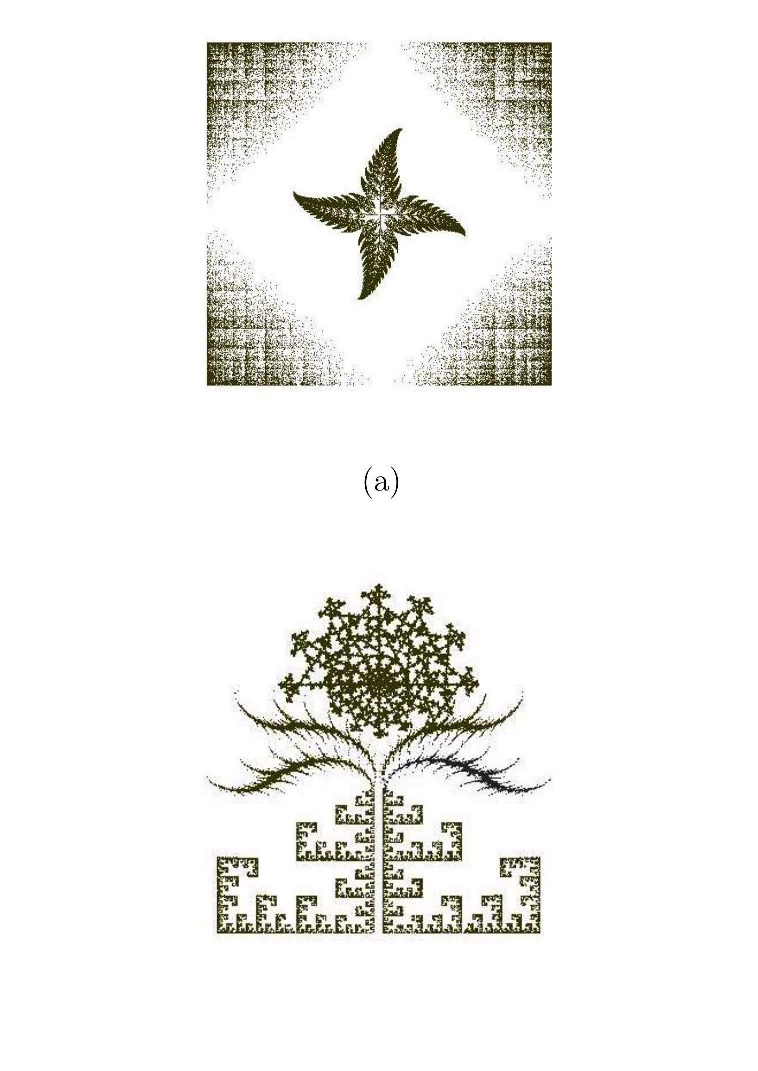


Figure 12: An example of composite transformations for Fern.



(a)

(b)

Figure 13: Two examples of the synthesized image frames that are integrated from multiple fractal sets.