Image Effects Using Contractive Mappings

JONAS GOMES
CICERO MOTA
ROMILDO JOSÉ DA SILVA
LUIZ VELHO

IMPA-Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110
22460-320 – Rio de Janeiro, RJ, Brasil
{jonas | mota | rjs | lvelho}@visgraf.impa.br

Abstract. In this paper we exploit the power of contractive mappings to create special image effects. Under this framework, images are represented as the attractor of an Iterated Function System (IFS) and can be reconstructed using Fractal Interpolation. By controlling parameters of the process, we obtain a wide range of image effects.

Keywords: Image effects; contraction mappings; fractal image compression.

1 Introduction

Special effects with images are very important in different fields. In particular we could mention the film and video industry and the digital publishing market. There is a great number of commercial software that can be used to obtain different image effects such as Adobe Photoshop, Fractal Paint, Corel Paint, Kaos Power Tools, etc.

Also, there is a vast literature covering the subject of special effects with images. These techniques range from the use of linear filters to non-linear ones, such as warping filters.

In this paper we introduce a new technique for obtaining special effects using an image codec based on contraction mappings of the plane. The image is encoded, and its reconstruction is parameterized in such a way that, by changing conveniently the parameters, we obtain different effects on the reconstructed image. This process allows us to produce visual results that are hard to achieve by other means. Moreover, it is a general method for creating image effects based on patterns and textures.

1.1 Related Work

Contraction mappings of the plane have been used for quite a while to obtain image compression. In this context the technique is called *fractal image compression*. The results are described in Barnsley and Hurd [2] and Barnsley and Jacquin [3]. A very good explanation of the techniques can be found in Fisher [4], where you can find details of the main existing algorithms.

This is the first work to use the powerful technique of contraction mapping encoding to obtain special effects with images.

1.2 Overview

The paper is organized as follows: Section 2 studies contraction mappings, introduces the concept of partitioned iterated function systems *(PIFS)* and presents the *contraction mapping theorem* which is the basis for the encoding theory using contractive mappings. Section 3 introduces a technique to encode and decode grayscale images using contraction mappings. Section 4 shows how to apply the codec introduced in Section 3 to obtain special effects with images. Section 5 gives several examples, and Section 6 comments on future work.

2 Contractive Mappings

A *contraction* is an application that decreases distances. More precisely, a mapping $F : A \longrightarrow B$ is a contraction when $d(F(x), F(y)) \leq sd(x, y)$, with $0 \leq s < 1$. The number s is called a *contractivity factor* for F.

We will illustrate the definition above with one example. Let F be defined on the unit square and given by $F(x, y) = (\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{4}) = \frac{1}{2}I + (\frac{1}{4}, \frac{1}{4})$, where I is the identity map of the plane. Then F is a contraction with contractivity factor $s = 1/2$, where distance on the plane is measured in the usual way (euclidean distance). Figure 1 shows the unit square transformed by the function F.

A very important property of a contraction $F: A \rightarrow A$ from a set A to itself is the existence of a unique point that remains fixed. That is, there exists a unique point $p \in A$ such that $F(p) = p$. The point p is called the *fixed point* of F. In the above example $p = (1/2, 1/2)$. The existence of a fixed point is a classical result stated precisely in the theorem below:

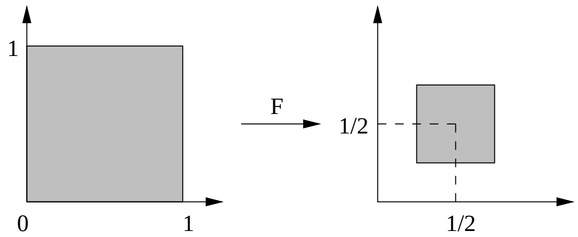


Figure 1:

Theorem 1 (Contraction Mapping Theorem) *Let $F : A \to A$ be a contraction on a complete metric space. Then F has a unique fixed point p. Furthermore, for any $x \in A$, the sequence $(x, F(x), F^2(x), F^3(x), \ldots)$ of iterates of x by F, converges to p.*

Notice that the theorem guarantees the existence, and also gives a method to compute the fixed point. For an arbitrary point $x \in A$ the sequence of iterates of x by F, $(x, F(x), F^2(x), F^3(x), \ldots)$, is called the *orbit* of x by F. The contraction mapping theorem says that the orbit of any point converges to the fixed point p.

2.1 Iterated Function Systems

In this section, we will introduce a distance on a collection of plane sets and show how to apply the contraction mapping theorem to encode any element from this collection. We will consider only bounded and closed subsets of the plane.

Let $\mathcal{B}$ be the collection of plane sets, we want to define a metric on $\mathcal{B}$ with perceptual characteristics, that is two sets are close to each other, whenever they are, perceptually, almost indistinguishable. This is illustrated in figure 2 for two sets of distance ε. The precise definition is given below.

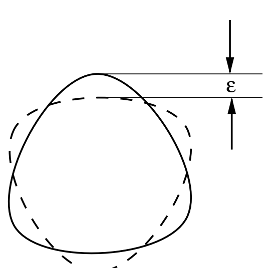


Figure 2:

Let A and B be sets in $\mathcal{B}$, the distance between them is defined by

$$d(A, B) = \max\{[A, B], [B, A]\},$$

where

$$\begin{aligned}
[A, B] &= \max\{d(x, B)\ ;\ x \in A\}, \quad \text{and} \\
[B, A] &= \max\{d(x, A)\ ;\ x \in B\}.
\end{aligned}$$

This distance turns the collection of plane sets into a complete metric space.

Suppose we have n maps $F_1, \ldots, F_n$ on the plane. We can define a map $F : \mathcal{B} \longrightarrow \mathcal{B}$ by

$$F(A) = F_1(A) \cup \cdots \cup F_n(A), \quad A \in \mathcal{B}.$$

It is possible to prove, see [4], that if each map F_i is a contraction, the map F is also a contraction on $\mathcal{B}$. Moreover if the contraction factor of each F_i is s_i, the contraction of F is given by the maximum value among the s_i's. It follows from the contraction mapping theorem, that there exists a unique set $K \in \mathcal{B}$ such that $F(K) = K$. The map $F\colon \mathcal{B} \to \mathcal{B}$ is called an *Iterated Function System (IFS)*. The fixed point of F in $\mathcal{B}$ is called the *attractor* for the *IFS*.

As an example, consider three contractions

$$\begin{aligned}
F_1(x, y) &= (\frac{x}{2}, \frac{y}{2}) = \frac{1}{2}I; \\
F_2(x, y) &= (\frac{x}{2} + \frac{1}{2}, \frac{y}{2}) = \frac{1}{2}I + (\frac{1}{2}, 0); \\
F_3(x, y) &= (\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{2}) = \frac{1}{2}I + (\frac{1}{4}, \frac{1}{2}),
\end{aligned}$$

and let A be the unit square $A = \{(x, y)\ ;\ 0 \leq |x| \leq 1 \text{ and } 0 \leq |y| \leq 1\}$. Figure 3 shows the action of F on the A.

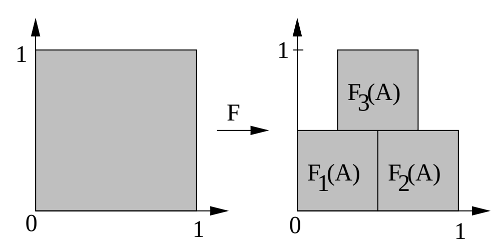


Figure 3:

Figure 4 shows some iterates of the sequence $F^n(A)$. This sequence converges to the set known as the *Sierpinski Triangle*. From the contraction mapping theorem, we obtain the Sierpinsky triangle independent of the starting set A.

In resume, the Sierpinsky triangle is reconstructed from the IFS $F = \{F_1, F_2, F_3\}$ using an arbitrary set as the starting point to iterate. This means that we can use the IFS as an encoding of the Sierpinsky triangle. A very interesting question can be posed now: *is it possible to*

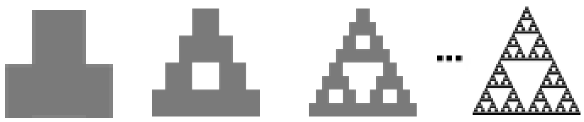


Figure 4:

use IFS's to obtain a similar encoding for an arbitrary set of the plane?

To state a solution to this problem, we introduce an affine IFS. That is, an IFS $F = \bigcup_i F_i$ such that each contraction F_i is defined by an affine mapping of the plane:

$$F_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}$$

The action of an affine mapping consists of rotation, reflection, translation, shearing and scaling. Therefore the fixed point of an affine IFS has a distinguished feature: it is composed of scaled version of itself, possibly modified by some of the warpings defined by some plane affine mapping. That is, it has a fractal affine self-similar structure. Of course, not every plane set has such a geometric fractal structure, but it is intuitively convincing that by subdividing an arbitrary set of the plane, we find many pieces that are approximately equal, up to an affine plane transformation. This is illustrated for the boundary of the set shown in Figure 5.

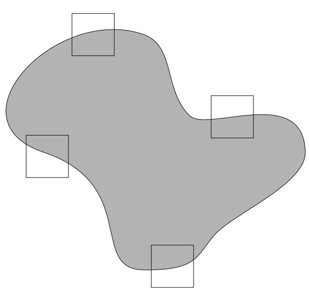


Figure 5:

Therefore, it is reasonable to expect that by conveniently subdividing a plane set, we are able to describe it as the fixed point of some affine IFS. This result is known as the *collage theorem*. A precise statement of it, and a proof, can be found in [1].

3 Images and Contractive Mappings

In this section we will show how to apply the encoding method described in Section 2 to obtain a representation for grayscale images.

3.1 Image Space

A grayscale image is defined by function $f : U \longrightarrow \boldsymbol{R}$ where U is a subset of the plane, called the support of the image. The value $f(x, y)$ represents the grey level at the point $(x, y) \in U$. In most cases the set U will be a rectangle.

A binary image is one where the image function f assumes only values 0 or 1. In this case, f defines a subset of the plane, characterized by the points of the plane where f assumes value 0 (black points). This remark allows us to deal with binary images as plane sets. In particular, the encoding technique for plane sets described in the previous section can be used to encode binary images.

Non-binary images are not perceptually identified with plane sets, and we must find a different way to measure the distance between two images, in order to encode them using contractions. This is done by identifying the image f with its graph

$$G(f) = \{(x, y, f(x, y))\ ;\ (x, y) \in U\}.$$

We must find a distance between two image graphs with perceptual characteristics: close images should look similar. Since the eye perceives intensity levels by an averaging process, a reasonable choice is to use the average metric

$$d(f, g) = \left(\int_U |f(x, y) - g(x, y)|^2 dxdy \right)^{1/2}.$$

With this metric the Contraction Mapping Theorem applies to the space of grayscale images.

3.2 Representation by Contractive Maps

As in the previous section we will look for a sufficiently great number of attractors pertaining to the grayscale image set, in such a way we can replace an original image by one of these attractors, with a minimal perceptual lost. By identifying a grayscale image $f : U \subset \boldsymbol{R}^2 \to \boldsymbol{R}$ with its graph, we define affine maps on the "graph space", by

$$W_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix}$$

The family W_i of affine mappings is comprised by two families of affine maps. A family F_i of plane affine mappings

$$F_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}$$

and a one-dimensional family of affine mappings

$$z \mapsto s_i z + o_i, \quad z \in \boldsymbol{R}.$$

The family F_i acts on the image domain, and the $1D$-family acts on the gray levels of the images. s_i scales the image gray levels, and it is called *contrast*. The parameter o_i adds a constant amount of gray to the image values, and it is called *brightness* parameter.

Let f be a image over U and $R_1, \cdots, R_n$ a partition of U such that there are subsets $D_1, \cdots, D_n$ in U satisfying $F_i(D_i) = R_i$ for each i (see figure 6). Define

$$W(f) := W_1(f|D_1) \cup \cdots \cup W_n(f|D_n),$$

where

$$W_i(f|D_i) = \{W_i(x, y, f(x, y)); (x, y) \in D_i\}.$$

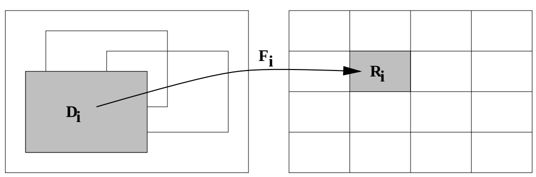


Figure 6:

W defines a mapping from the space of images onto itself. If we choose the F_i to be contractions and the $s_i \leq 1$, then we can prove that W is also a contraction. Therefore, it has a fixed point f_W.

The contraction mappings W_i between pieces of U with the above properties constitutes what is called a *Partitioned Iterated Function System (PIFS)*. The sets D_i are called *domains* and the sets R_i are called *ranges*.

Now, again, we want to solve the inverse problem, that is, given an image f over U we hope to find a *PIFS* that has f as its fixed point. In practice, we look for *PIFS* with a fixed point f_W close to f. So we can encode an image f by storing the contraction $W = \bigcup_i W_i$, the domains D_i and the associated ranges R_i. For decoding, one starts with an arbitrary initial image g and apply W repeatedly to g until we get a sufficiently good approximation of f_W.

In resume, the problem is to find the sets R_i, the corresponding sets D_i and the transformations W_i. In the next section we will present a method to accomplish this.

3.3 Computing the Representation

Suppose we have a 256×256 grayscale image f. Consider a regular grid of 8×8 non-overlapping squares. This grid creates a partition $R_1, R_2, \ldots, R_{1024}$ of the image domain U. Let $\boldsymbol{D}$ be the collection of all 16×16 subsquares D_j of the image domain (see figure 6). In order to find a *PIFS* to encode f, it is necessary to find, for each R_i, a set D_i in $\boldsymbol{D}$ such that $f|D_i$ is similar to $f|R_i$ by an affine transform W_i.

Given a set D_i int $\boldsymbol{D}$ there are eight ways to map it over R_i. Moreover the domain D_i is four times the size of any range set R_i. Therefore, we must subsample $f|D_i$. In conclusion, we have to make 464648 comparisons in order to choose the D_i in $\boldsymbol{D}$ that minimizes the distance

$$d\big(W_o(f|D), f|R_i\big)$$

where W_o is an affine domain transformation which leaves the gray values unaltered. That is, a transformation of the type

$$W_o \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ 0 \end{pmatrix}.$$

This enables us to choose the coefficients a_i, b_i, c_i, d_i, e_i and f_i corresponding to the best domain transform W_i. After we do this for each range R_i we need to find optimal values for the contrast s_i and the brightness o_i to obtain the best transform W_i. It is a easy task if we are using the RMS metric. Suppose we have the sets D_i and R_i with respective pixel intensities given by $d_1, \cdots, d_{64}$ (after subsampling) and $r_1, \cdots, r_{64}$. In order to obtain the best s_i and o_i for W_i we have to minimize the error

$$E(o, s) = \sum_{i=1}^{64} (s \cdot d_i + o - r_i)^2.$$

We reach the minimum of E when its partial derivatives are zero and it happens when

$$s = \frac{\left[64 \cdot \sum_{i=1}^{64} d_i r_i - \sum_{i=1}^{64} d_i \sum_{i=1}^{64} r_i\right]}{\left[64 \cdot \sum_{i=1}^{64} d_i^2 - \left(\sum_{i=1}^{64} d_i\right)^2\right]}$$

and

$$o = \frac{1}{64}\left[\sum_{i=1}^{64} r_i - s \cdot \sum_{i=1}^{64} d_i\right]$$

4 Image Reconstruction and Special Effects

The *PIFS* representation is commonly used for image compression. In such an application, the original image is reconstructed from a compact description with minimal distortion.

However, this representation could also be used for creating special effects. If the parameters of the process are carefully manipulated, we may alter the reconstructed image, introducing desired visual features into it. Note that, in this case, the reconstructed image will be different from the original encoded image.

In the rest of this section, we describe how the computational machinery of contractive mappings can be controlled to produce these special effects.

4.1 Fractal Interpolation

The scheme for encoding and decoding an image using *PIFS* is essentially a process of sampling and reconstruction (interpolation). The sampling process is based on the principle that if a region of the image is sufficiently small then it is a scaled copy of itself up to an affine transformation. We subdivide the image in regions and associate to each region an affine contraction W_i. The value o_i is greatly influenced by the mean intensity of the range R_i, so the set $\{o_1, \ldots, o_n\}$ is roughly a subsampled copy of the image. The contractions have the information necessary to increase the resolution. By iterating the affine mappings we are reconstructing the details from coarse information. This process is called *affine fractal interpolation*. It is illustrated in Figure 7 for $1D$-images (i.e. a scan-line of an image).

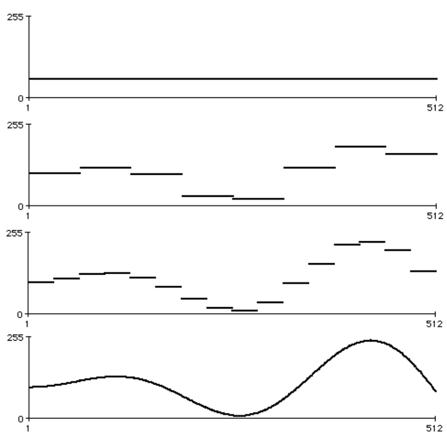


Figure 7:

Note that, the iteration process produces a sequence which starts with an arbitrary image and converges to the encoded image. Most importantly, the intermediate images in this sequence are interpolations "in a fractal sense" between the initial and final images. Therefore, by selecting one of these iterates we obtain an image that combines features of both images according to its index in the sequence.

4.2 Parameters

The reconstruction method used in the decoding phase of the *PIFS* codec depends on several parameters. The main parameters of this process are:

- The initial image;
- The number of iterations.
- The size of the range-domain pairs.

As we have seen in the previous subsection, if the iteration process is carried out just a few times, we obtain an image that resembles the encoded image but also incorporates features of the initial image.

The above parameters control the visual effects. The choice of initial image is perhaps the most important parameter, since it defines the kind of features that will be incorporated into the image. Here we will use regular patterns and textures.

The number of iterations indicates the interpolation level, and therefore the strength of the initial image features.

The size of the ranges influences the feature granularity, while the ratio of range/domain sizes gives the contractivity factor. Here the size of the domains will be twice the size of the ranges.

4.3 Algorithm

The whole process is constituted, essentially, by the following steps:

1. choice of parameters to encode the image;
2. image encoding;
3. selection of the initial image;
4. choice of parameters to decode the image; and
5. image decoding.

The proper combination of the choices made in 1, 2, and 3 are the key to achieve a desired effect in the reconstructed image.

5 Examples

In this section we will illustrate the use of contraction mappings to create image effects. We will show several examples generated by applying the method of Section 4 to test images. We will also analyze the relationship between the various parameters in the process and its influence in the final image.

5.1 Test Images and Patterns

We have selected two test images, Lenna and Island, that are shown respectively in Figure 8 (a) and (b). They are 8 bit grayscale images with resolution of 512×512 pixels. These images depict the most common subjects of real world applications: a portrait and a natural landscape.

The initial image for the fractal interpolation will be composed by a regular pattern. The geometry and contrast of the pattern will greatly influence the effect obtained. We will use, for our examples, the patterns that appear in Figure 9 (a) and (b). The first is formed by squares and the second by stripes. Each type of pattern

(a) (b)

Figure 8: Images of (a) Lenna and (b) Island.

is defined by its own construction parameters. For the square pattern, the parameters are the size of squares and the distance between them. For the stripe pattern, the parameters are the distance between the stripes and their slope.

(a) (b)

Figure 9: Patterns - (a) Square and (b) Stripe (b)

We will see that the relationship between pattern resolution and domain size is very important in the final effect. Pattern elements in the image are clipped, scaled and copied to a new location at each iteration. Therefore, the resolution of the pattern should be roughly the same as the domain size.

5.2 Results

We now present a series of examples demonstrating the influence of the number of iterations in the final effect.

Figure 10 is the first iteration of the reconstruction sequence for Lena. The initial image is a pattern formed by 12×12 black squares distancing 4 pixels from each other. The domain size is 16×16 pixels.

Figure 11 is the second iteration of the reconstruction for the Island. The initial image is a 45 degree stripe pattern with width and separation of 10 pixels. The domain size is 16×16 pixels.

Figure 12 is the third iteration of the reconstruction sequence for Lena. The initial image is formed by a stripe pattern with width and separation of 10 pixels. The domain size is 16×16 pixels. Note that, although we cannot see the pattern itself, a subtle texturing effect is obtained.

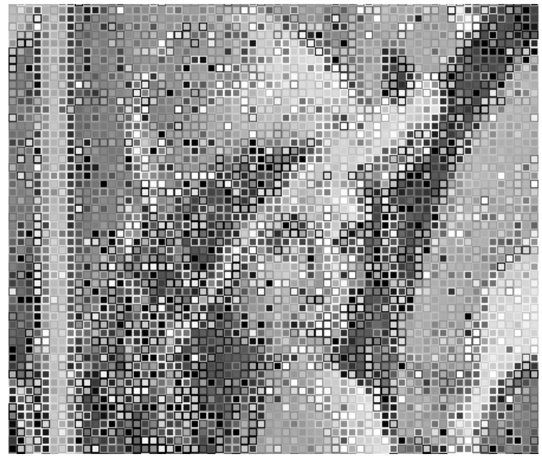


Figure 10: First iteration with square pattern.

Figure 11: Second iteration with stripe pattern.

Figure 12: Third iteration, and detail

This is more clearly shown in the enlarged fragment of the eye.

Figure 13 shows two fragments of the fourth iteration of the reconstruction sequence for the Island image, using the same parameters of the second example. Here, the sequence already converged to its fixed point and, consequently, the final image is almost indistinguishable from the original. But, as can be seen, different parts have better convergence than others. The clouds are perfectly reconstructed, while the palm trees exhibit blocky artifacts. This is because the uniform image partition scheme does not adapt to local image characteristics.

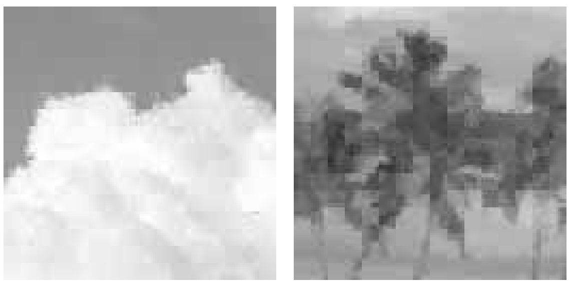


Figure 13: Two details of the fourth iteration for the Island

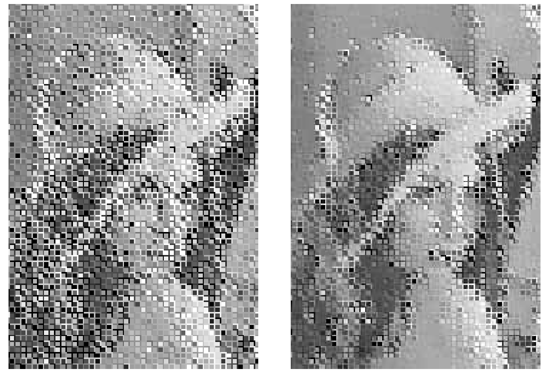
5.3 Analysis

In the following we present some examples that clarify the relationship between the parameters of the process. We begin investigating the interdependence of domain-range sizes, number of iterations and pattern contrast. Next, we analyze the effects of varying the ratio of pattern and domain sizes. We, then, discuss the influence of pattern intensity in the rate of convergence. Finally, we demonstrate the result of domain rotation on patterns that have directional features.

Figure 14(a) was obtained with the following parameters: one iteration, 16×16 domains, 8×8 ranges, and 16×16 black square patterns distant 4 pixels apart. A similar result was obtained in Figure 14(b) with parameters: two iterations, 32×32 domains, 16×16 ranges, and 32×32 black square patterns distant 8 apart. Observe that both images have the same kind of effect but, the first image has greater contrast, while the second has better intensity balance.

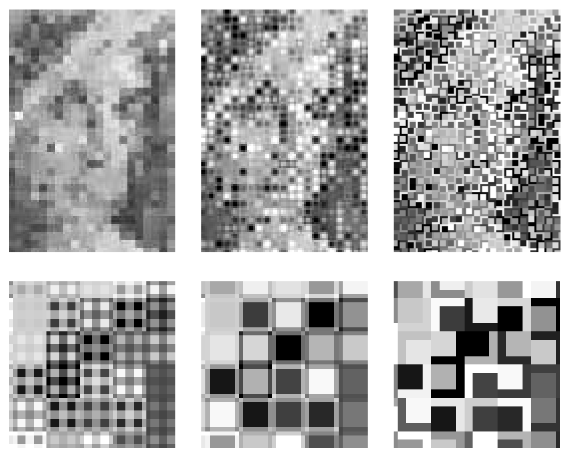
Figure 15 shows the different effects achieved by varying the pattern size with a fixed domain size of 16×16. In Figure 15(a) the pattern is half of the domain, in Figure 15(b) the pattern and domain have the same size, and in Figure 15(c) the pattern is twice the domain size.

Figure 16 illustrates the effect of the pattern mean intensity over the final image. The mean intensity of Lena image is 123. The images in Figures 16(a), (b) and (c), were obtained in one iteration, with pattern mean intensity respectively equal to 59, 123, and 157.



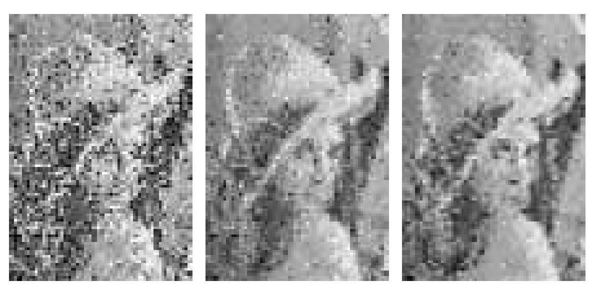
(a) (b)

Figure 14: Comparison - number of iterations.



(a) (b) (c)

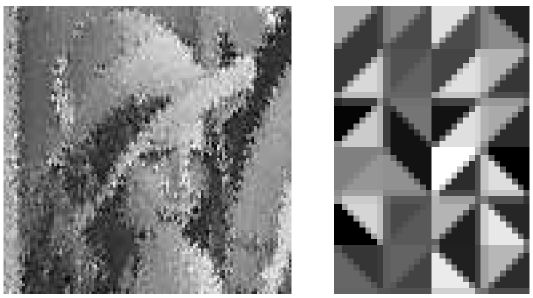
Figure 15: Comparison - size of the patterns.



(a) (b) (c)

Figure 16: Comparison - initial image intensities.

Figure 17 shows the effect of rotation transforms over domains. The image in Figure 17(a) was generated with two iterations, and 16×16 ranges. The initial image was a stripe pattern with width and separation of 32 pixels. The detail in Figure 17(b) is an enlarged piece of the image that exhibits the diagonal pattern in various orientations.



(a) (b)

Figure 17: Effects of domain rotation.

6 Conclusions

In this paper we have presented a new method to generate image effects using contractive mappings over image spaces. This method employs an image encoding / reconstruction scheme based on Partitioned Iterated Function Systems (*PIFS*), which represents the image as the fixed point of a set of contraction transformations. In such a scheme, encoding amounts to analyzing image similarities and reconstruction consists of applying transformations iteratively to an arbitrary image until it converges to the encoded image.

We have shown that the above reconstruction procedure can be seen as an affine fractal interpolation between the initial and final images. Our method is based on the observation that by selecting an appropriate initial image and controlling the interpolation process, it is possible to create powerful image effects. Moreover, we have illustrated, through several examples, that a wide variety of visual features can be obtained with our method, ranging from rich graphic patterns to very subtle textures. We also have discussed how to control the right combination of parameters in order to generate a desired image effect.

6.1 Current Work

Current work is going into several directions, both to enhance the method, as well as, to turn it into a professional image processing tool.

First, we intend to adopt a quadtree-based image partition scheme. This is a more general decomposition, which has as a special case the uniform partition that we currently use. The main advantage of employing this decomposition is that it produces effects adapted to the image characteristics. Figure 18 shows the result of an early experiment with quadtree decomposition.

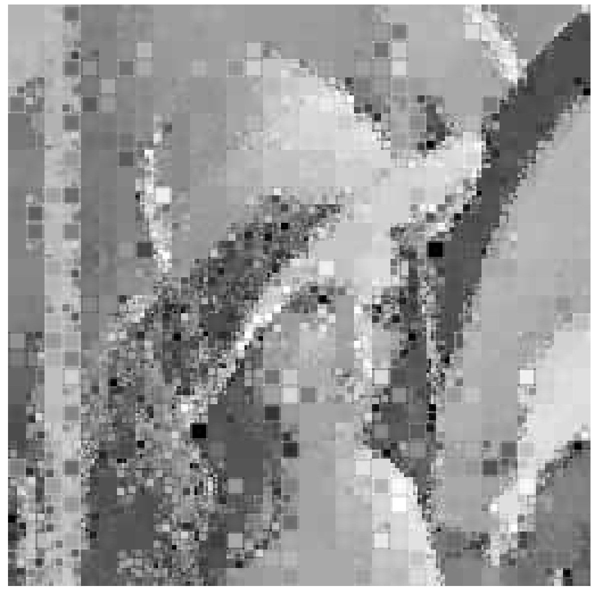


Figure 18: Adaptive image partition

Second, we would like to extend our method to color images. This is a straightforward extension of our present algorithm, which works only with gray scale images. In terms of image representation, we will have to deal with a color space, such as RGB, HSV, or IYQ. In terms of the effect generation, we will have more parameters to control. In particular, it would be interesting to employ color patterns derived from the palette of image colors.

Third, we intend to implement this method as a Photoshop plug-in, making it widely available as a production tool for graphic artists. Important issues in that respect are: efficient processing and interface design.

7 References

[1] M. F. Barnsley. Fractals Everywhere. Second Edition. Academic Press Professional.

[2] M. F. Barnsley and L. P. Hurd. Fractal Image Compression. AK Peters, Ltd., Wellesley, Ma., December 1992.

[3] M. F. Barnsley and A. Jacquin. Applications of Recurrent Iterated Function to Images. SPIE Vis. Comm. Image Proc., 1001:112-131, 1988.

[4] Y. Fisher. Fractal Image Compression. Springer-Verlag, 1995.