Proposal of an Extension Pre-trained Models of Image Recognition Based on FractalDB

Hiroki Saito*

February 19, 2023

Abstract

Image recognition tasks using deep learning often use pre-trained models. Most of the datasets used for such pre-training are large natural image datasets, but there is another type of dataset called a FractalDB that generates a large amount of fractal image data and is also automatically annotated. Although the accuracy of pre-training with this dataset is known to be relatively better than with scratch and other self-supervised learning methods, we believe that the accuracy could be further improved if the variety and data scalability of the fractal image dataset could be further enhanced. Therefore, we propose several methods of image augmentation and image generation to maximize their performance. The results show that there is no significant difference in the learning accuracy of fine tuning, but the improved image generation method achieves slightly higher accuracy than the present method under the same conditions. Therefore, there is a high potential for further improvement by constructing higher-scale datasets and improving the parameters of the proposed method.

1 Introduction

Today, pre-training and fine-tuning are considered essential methods for solving image recognition tasks. Pre-training using massive images, from one million to nearly ten million, such as ImageNet[3] and Places[10], has become the standard. In this situation, a dataset composed of many fractal images, called FractalDB, has been proposed[5]. Experiments with FractalDB have shown that the accuracy of fine-tuning performed on pre-trained models using this dataset is better than the scratch and other self-supervised learning models. Another advantage besides the high accuracy is that FractalDB differs from conventional datasets, which require natural images and manual labeling, thus creating a more accurate and efficient large-scale dataset.

We generate the fractal images in Fractal DB using the IFS method, in which we plot dots on an image canvas by repeated affine transformations. Although this method can generate many fractal images with unique shapes, certain parameter combinations can produce fractal images that are featureless or similar to other fractal images. In this case, the pre-trained model may need help to learn fractal shape patterns well during pre-training. As a result, the accuracy in fine-tuning remains the same. Thus, this paper proposes a method for generating various fractal images, not only by linear transformations but also by nonlinear ones. This method will increase the number of fractal shape patterns and enable more effective learning by learning various shapes. Also, the additional nonlinear transformations increase the number of different fractal images to be generated, allowing us to construct higher-scale datasets than before. This method would add new images to the conventional FractalDB to build a more effective fractal image dataset.

In addition, we propose the Gaussian Blur as a data augmentation for FractalDB. It is because the

* Department of Frontier Media Science (FMS), School of Interdisciplinary Mathematical Sciences, Meiji University

blurring process does not significantly destroy the shape of the fractals and thus can increase the image data without reducing the quality of the training. The blurring process can also generate grayscale images from black-and-white binary pictures. It will allow us to reproduce some of the seamless tonal changes in natural images of shapes and thus enable us to learn more effectively. We verified the effectiveness of the proposed methods by comparing the accuracy of fine-tuning on the dataset presented in this paper, ImageNet, and FractalDB.

2 Related work

2.1 Pre-training with a dataset composed of fractal.

We know it is adequate for pre-training to use a large dataset consisting of nearly 10 million images, such as ImageNet. In ImageNet, those large numbers of images are those of natural objects, and humans entirely perform the labeling process. Here we have troubles such as image privacy, gender issues, and labeling errors affecting pre-training. Fractal DB has the potential to solve these problems and still achieve higher accuracy than Scratch or other self-supervised learning. This dataset is unique because it does not use natural images and performs all labeling automatically; image generation utilizes an algorithm based on a mathematical method called IFS. It allows effective pre-training while solving the problems associated with using natural images. In addition, image generation by affine transformations of IFS and labeling by their parameters can construct large image datasets without human help. We propose a method to increase accuracy by focusing on the scalability and variety of FractalDB.

2.2 Improving Fractal Pre-training.

It is well-known that the fine-tuning accuracy by pre-training in FractalDB can be improved by multi-instance prediction and practical selection of fractals using Singular Value Decomposition(SVD)[1]. Multi-instance prediction is a pre-training classification method initially proposed by [1]. This method can efficiently learn the attributes of a fractal by containing multiple instances of the fractal in a single image. The second method is concerned with sampling fractals generated by IFS codes. A technique using parameters obtained by SVD and an SVM classifier efficiently allows us to sample geometrically preferable fractals from IFS codes. Since FractalDB uses only affine transformations to build fractals, we can quickly introduce SVD to analyze the fractal shapes mathematically. And the SVM classifier allows us to classify fractals geometrically proper or not; we efficiently eliminate images if they are too simple or too scattered.

2.3 Generation of various fractal images with IFS.

Fractal images generated by Iterated Function System (IFS) have a wide variety of shapes depending on variation transformations and parameters. They also allow us various data processing methods, such as coloring and motion blur[8]. By applying these extensions of IFS to FractalDB, we can further improve the accuracy. FractalDB uses only basic methods; thus, we have much more scope to generate various fractal images.

3 Methods

In this short paper, we propose a new dataset based on FractalDB to improve the accuracy of FractalDB through an extended fractal image generation method and an image transformation process for data augmentation. In fractal image generation, using nonlinear transformations, we can produce unique and distinctive fractal images in the same process of IFS[8]. The appendix shows the nonlinear function formulas and images used in this experiment. As an example, the formulas for a linear function and several nonlinear functions are shown here.The variables used in the formula are as follows

$$r = \sqrt{x^2 + y^2}$$
$$\theta = arctan(x/y)$$

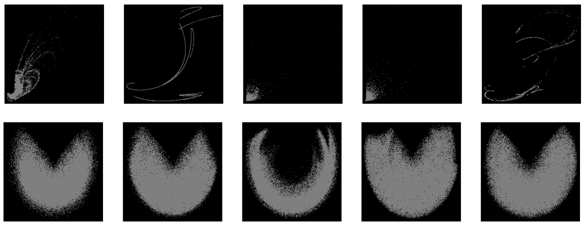
The mathematical formula for linear and some nonlinear function are:

$$V_{linear}(x, y) = (x, y)$$
$$V_{sinusoidal}(x, y) = (sin\, x, sin\, y)$$
$$V_{spherical}(x, y) = \frac{1}{r^2} \cdot (x, y)$$
$$V_{hand\_kerchief}(x, y) = r \cdot (sin(\theta + r), cos(\theta - r))$$
$$V_{power}(x, y) = r^{sin\theta} \cdot (cos\theta, sin\theta)$$

Figure1: Examples of images that can be generated by IFS using several nonlinear functions. identity function is a)linear and some nonlinear functions are b)sinusoidal, then c)spherical d)hand_kerchief and finally e)power

Here, we excluded some images from the viewpoint of effective pre-training after introducing several types of nonlinear transformations. The exclusion criteria and some images generated by them are as follows

- If the number of plotted points is small, that is, less than a certain threshold, in the entire image, we remove the nonlinear transformation from IFS. Because if the image with few plotted points, we cannot characterize or classify it, and hence we cannot expect efficient learning.
- If the image view does not change much after adding nonlinear transformations, we remove the nonlinear transformation from IFS. In this case, we cannot expect a rich variation of fractal images, which may harm the classification in the pre-training.



a

b

Figure2: Examples of images excluded in this experiment. Image a has fewer points to plot. Image b shows little change in fractal shape.

Following these criteria in selecting transformations, we use 17 types of transformations in IFS, consisting of one linear transformation and 14 different non-linear transformations. We expect a dataset using these transformations to give us a more effective pre-training model than FractalDB. In addition, we present Gaussian Blur as an additional augmentation in the dataset of images. Gaussian Blur augments the diffusion of drawing dots by random numbers. Figure 1 shows that the larger the variance value, the more the image loses sharpness, and the more blur is gradually applied. If the variance is not zero, the output image is grayscale, so we expect these blued fractal images to be helpful in the pre-trained process.

Figure3: Data Augmentation with Gaussian Blur

4 Experiments

To demonstrate the effectiveness of the datasets proposed in this paper, we used resnet50 for training[4]. As for the datasets for fine-tuning, we use five datasets: CIFAR-10, CIFAR-100, Stanford Cars, Flower-102, and Food-101[9, 6, 7, 2]. We compare the results with trained models from scratch, ImageNet, and nonlinear and gaussian datasets. The number of categories in the fractal DB for the pre-training was all unified to 1000. In Table 1, we show data on C10 and C100. Here, the accuracy was below that of Scratch training in both the nonlinear and gaussian cases. The results of the accuracies are almost similar for the other datasets. However, we can confirm that nonlinear is slightly more accurate than linear, although it is less accurate than scratch. These show that constructing fractal datasets with nonlinear transformations is more effective in pre-training than in the linear case. Thus, the problem of low accuracy is likely to be caused by the instance augmentation method. Focusing on gaussian in C10 and C100, the accuracy is lower than that of points without the usual augmentation method.

	C10	C100	Cars	Flowers	Food
Scratch	90.97	71.30	40.48	33.76	73.85
ImageNet-1k	96.44	83.85	87.94	91.23	85.04
linear-point	88.76	68.13	50.98	40.72	68.91
linear-gaussian	88.25	62.93	35.62	30.27	35.92
nonlinear-point	90.12	69.77	61.77	53.33	72.51
nonlinear-gaussian	90.46	66.03	47.86	37.94	40.83

Table1: Classification accuracy when using Scratch, ImageNet, and fractal datasets. There are four types of fractal datasets when nonlinear transformations and Gaussian blurring are applied respectively({linear,nonlinear}-{point,gaussian}). There are five datasets used for classification: CIFAR-10(C10),CIFAR-100(C100),Stanford Cars(Cars),Flower-102(Flower),Food-101(Food)

5 Discussion

This experiment showed the effect of fractal datasets with nonlinear transformations. Gaussian Blur resulted in lower accuracy than a point, even though the number of images increases. We concluded that Gaussian Blur was ineffective in augmenting the data because the use of Gaussian Blur may have negatively affected the pre-training. The following two points are necessary for the benefit of nonlinear transformations to improve the accuracy further.

- **More variations of nonlinear transformations.**
  The number of kinds of transformations is currently 14, which is a few. To improve the accuracy, we need to generate various shapes by using much more numbers of nonlinear transformations.
- **Eliminate nonlinear transformations that negatively affect pre-training**

Through this experiment, we observed the effects of each nonlinear transformation from the viewpoint of simpleness and scatteredness of images because such transformations would harm learning. Due to this elimination, the accuracy of the nonlinear transformation case was slightly higher than that of the linear transformation case. This observation shows that it is necessary to eliminate transformations that generate images that are likely to have a negative impact.

Acknowledgement

I want to thank Professor Kazushi Ahara for his constructive suggestions and continuous support during research discussions.

References

[1] Connor Anderson and Ryan Farrell. Improving fractal pre-training, 2021.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. In *Asian Conference on Computer Vision (ACCV)*, 2020.

[6] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[7] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[8] Scott Draves Spotworks and Erik Reckase Berthoud. The fractal flame algorithm. 2008.

[9] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[10] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Appendix: Example of variation functions and images used for training

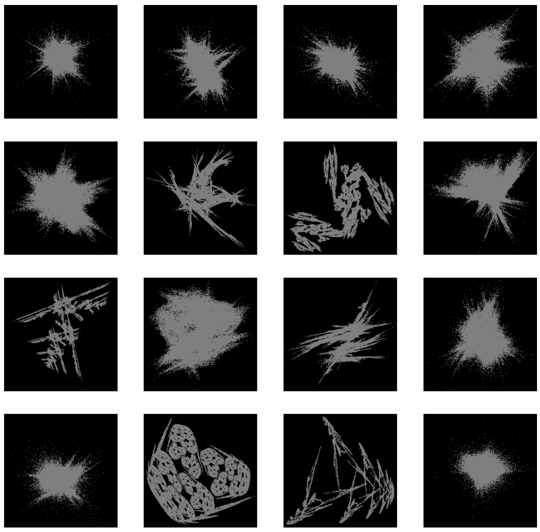
The mathematical formulas and some of the images generated by the linear and 14 different nonlinear functions used in this experiment are shown. For a given nonlinear function, 16 different categories of images are shown, and even fractal images with the same nonlinear function are able to produce a wide variety of images. The variables used in these formulas are:

$$r = \sqrt{x^2 + y^2}$$
$$\theta = arctan(x/y)$$

The mathematical formula and images for linear and some nonlinear function are as follows.

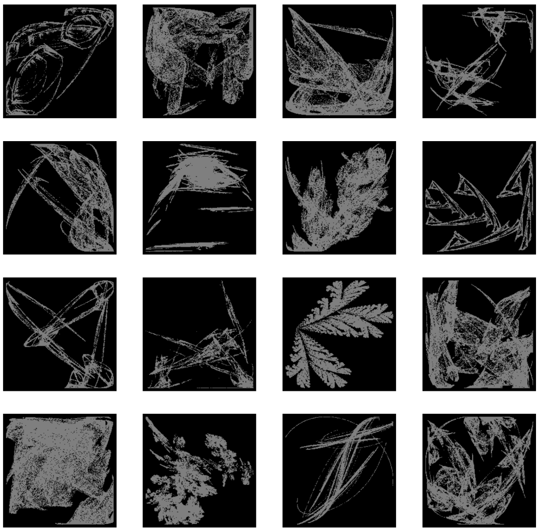
Linear

$$V_{linear}(x, y) = (x, y)$$



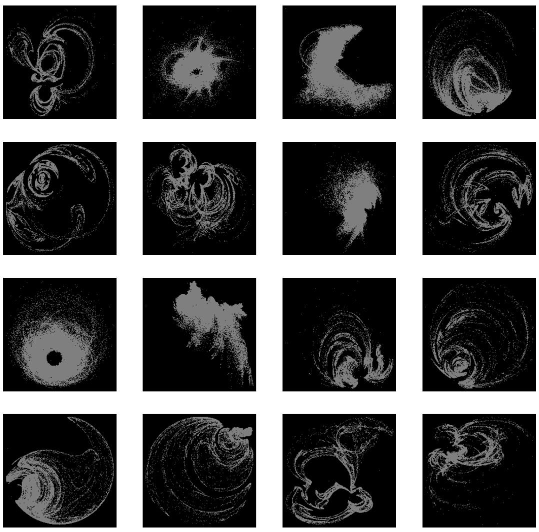
Sinusoidal

$$V_{sinusoidal}(x, y) = (sin\, x, sin\, y)$$



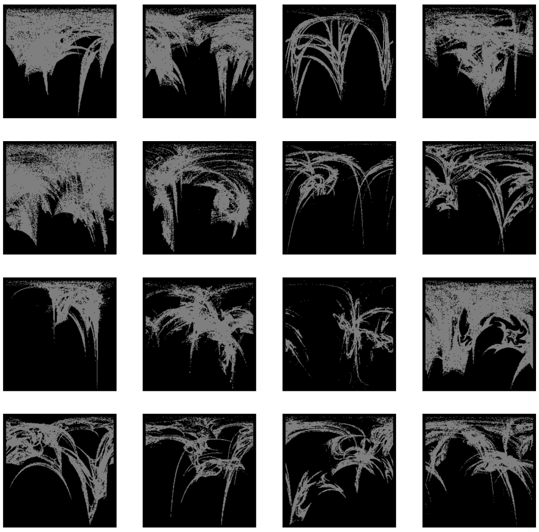
Spherical

$$V_{spherical}(x, y) = \frac{1}{r^2} \cdot (x, y)$$



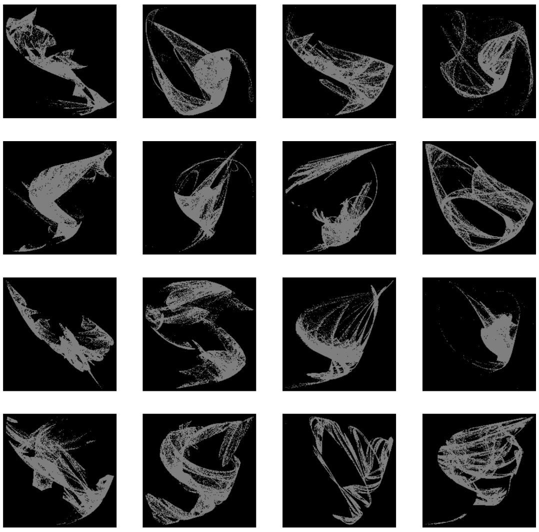
Polar

$$V_{polar}(x,y) = (\frac{\theta}{\pi}, r-1)$$



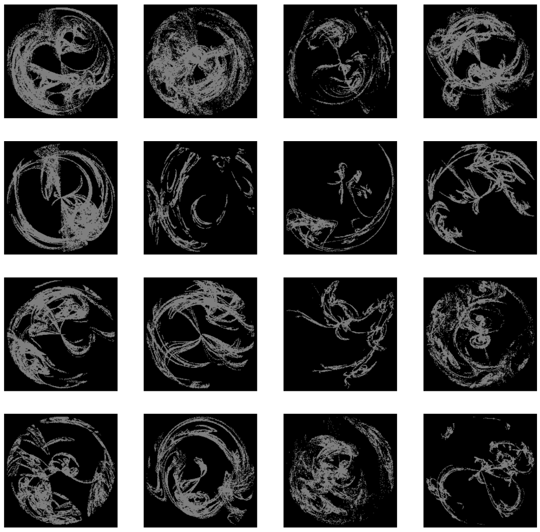
Hand_kerchief

$$V_{hand\_kerchief}(x,y) = r \cdot (sin(\theta + r), cos(\theta - r))$$



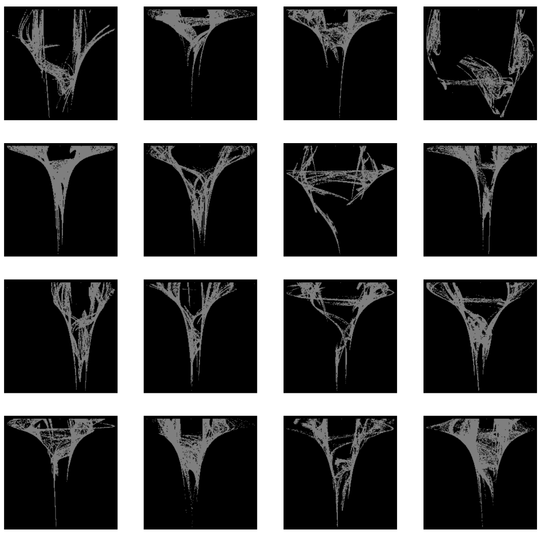
Disc

$$V_{disc}(x,y) = \frac{\theta}{\pi} \cdot (sin(\pi r), cos(\pi r))$$



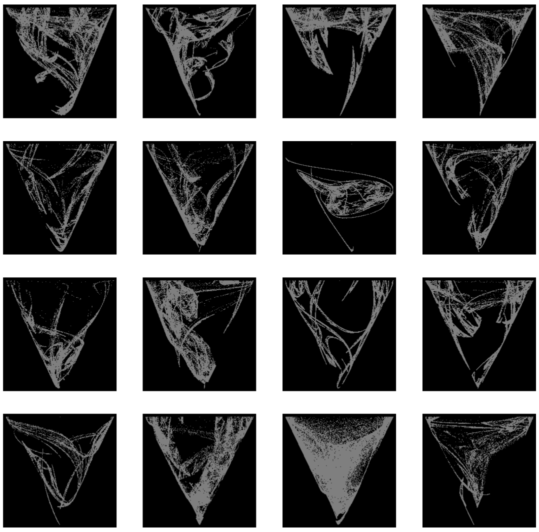
Hyperbolic

$$V_{hyperbolic}(x,y) = (\frac{sin(\theta)}{r}, rcos\theta)$$



Diamond

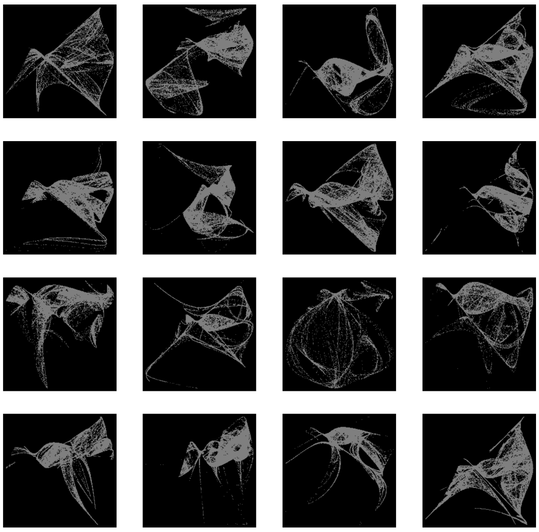
$$V_{diamond}(x, y) = (sin\theta cosr, cos\theta sinr)$$



Ex

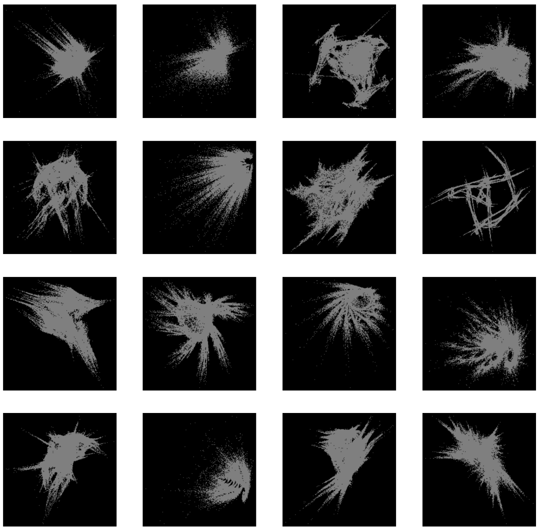
$$p_0 = sin(\theta + r), p_1 = cos(\theta - r)$$

$$V_{ex}(x, y) = r \cdot (p_0^3 + p_1^3, p_0^3 - p_1^3)$$



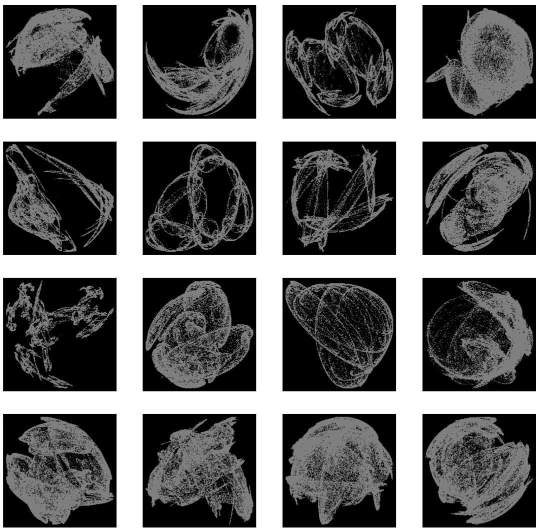
Bent

$$f(x) = \begin{cases} (x, y) & x \geq 0, y \geq 0 \\ (2x, y) & x < 0, y \geq 0 \\ (x, y/2) & x \geq 0, y < 0 \\ (2x, y/2) & x < 0, y < 0 \end{cases}$$



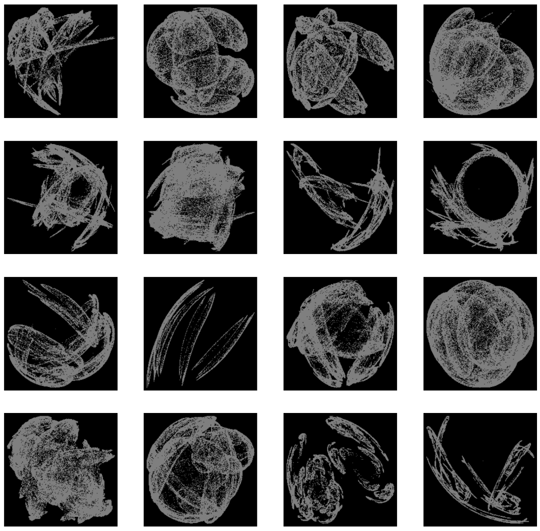
Fisheye

$$V_{fisheye}(x, y) = \frac{2}{r+1} \cdot (y, x)$$



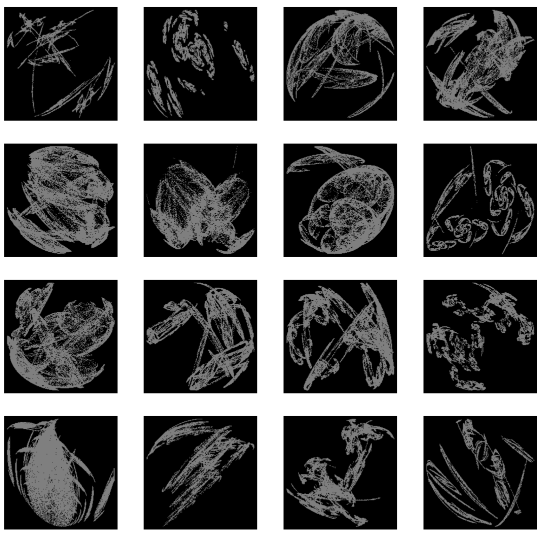
Eyefish

$$V_{eyefish}(x, y) = \frac{2}{r+1} \cdot (x, y)$$



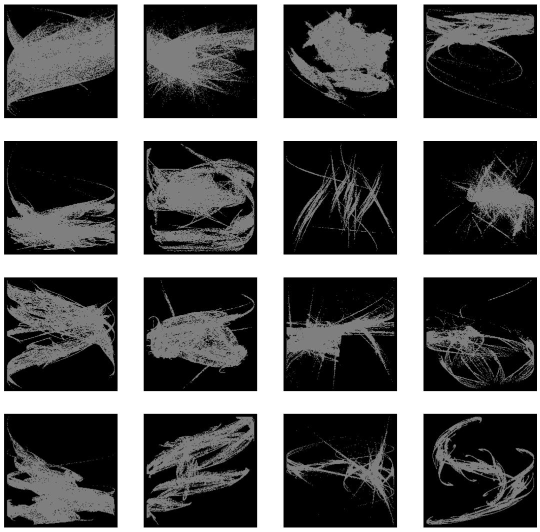
Bubble

$$V_{Bubble}(x, y) = \frac{4}{r^2+4} \cdot (x, y)$$



Cylinder

$$V_{cylinder}(x, y) = (sinx, y)$$



Power

$$V_{power}(x, y) = r^{sin\theta} \cdot (cos\theta, sin\theta)$$