Available online at www.sciencedirect.com
ScienceDirect
Chaos, Solitons and Fractals 32 (2007) 1429–1439
CHAOS SOLITONS & FRACTALS
www.elsevier.com/locate/chaos

Interpolation decoding method with variable parameters for fractal image compression

Chuan-jiang He *, Gao-ping Li, Xiao-na Shen

College of Mathematics and Physics, Chongqing University, Chongqing 400044, PR China

Accepted 15 November 2005

Communicated by Prof. M.S. El Naschie

Abstract

The interpolation fractal decoding method, which is introduced by [He C, Yang SX, Huang X. Progressive decoding method for fractal image compression. IEE Proc Vis Image Signal Process 2004;3:207–13], involves generating progressively the decoded image by means of an interpolation iterative procedure with a constant parameter. It is well-known that the majority of image details are added at the first steps of iterations in the conventional fractal decoding; hence the constant parameter for the interpolation decoding method must be set as a smaller value in order to achieve a better progressive decoding. However, it needs to take an extremely large number of iterations to converge. It is thus reasonable for some applications to slow down the iterative process at the first stages of decoding and then to accelerate it afterwards (e.g., at some iteration as we need). To achieve the goal, this paper proposed an interpolation decoding scheme with variable (iteration-dependent) parameters and proved the convergence of the decoding process mathematically. Experimental results demonstrate that the proposed scheme has really achieved the above-mentioned goal.

1. Introduction

Fractal image compression proposed by Barnsley, especially the practical compression scheme presented by Jacquin [1], has been attracting considerable attention of the researchers in the field of image compression over the past decade (e.g., [2–7]). The novelty and the simplicity of the fractal image compression have also received a lot of interests from the researchers in other areas of image processing (e.g., [8–11]).

Fractal image compression is based on the representation of an image by a contractive transformation, for which the reconstructed image is approximately its fixed point (image) and close to the original image. The parameters of the contractive transformation constitute the fractal code of the original image. The technique is mathematically based on the collage theorem and Banach's fixed-point theorem. Finding a contractive transformation with a given image as its fixed point is called the inverse problem in the context of fractal image compression, which is considerably difficult and challenging [2]. The collage theorem suggests a possibly sub-optimal method of solving the difficult problem; it inverts the

* Corresponding author.
E-mail address: chuanjianghe@sina.com (C.-j. He).

0960-0779/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.chaos.2005.11.058

problem into the one of finding a transformation mapping the original image close to itself, which is usually tractable computationally. Hence, in order to guarantee that the fixed point is close to the original image, one only needs to minimize the "collage" distance between the original image and its transformed version, rather than the distance between the original image and the fixed point of the transformation. The Banach's fixed-point theorem guarantees that the unique fixed point of the contractive transformation can be approximated to an arbitrary accuracy by repeating applications of the transformation on consecutive results, commencing with any initial image. The collage theorem guarantees that the decoded image, the approximate fixed point of the contractive transformation, is close to the original image.

The fast decoding is considered as one of the remarkable advantages for fractal image compression, in which the decoded image and the original image are usually very much alike at only the third iteration. Although the conventional fractal decoding is relatively fast, one may still wish to find faster decoding schemes for many applications where the decoding speed is very important (e.g., in real-time video). Therefore, most researchers interested in the fractal decoding have been focusing their attention to developing faster decoding methods in recent years (e.g., [12–15]).

On the other hand, the progressive fractal decoding is useful for low bandwidth transmission as done previously by Barnsley's Iterated Systems, Inc., which patented a lot of work with fractal image compression [16]. The progressive fractal decoding for low bandwidth transmission is one of the works done commercially by this company. However, the conventional fractal decoding method is not directly suitable for the case because it does not provide any control parameters for the decoding procedure; in other words, it cannot be controlled in a proper manner. Therefore, it is most desirable that the fractal decoding procedure could be controlled in some manners, which are suitable for progressive decoding. Besides, controlling decoding process might be useful for some multimedia applications such as making cartoons by computer-aided technologies [16], where the desired image might need to be gradually shown up from the other image in a progressive manner. The conventional fractal decoding, however, is also not suitable for this case for the same reason above. It is therefore interesting to consider other possible fractal decoding schemes, which are capable of illustrating the convergence of the iterative process in a different manner, or controlling the iterative process as we need, or displaying how image details are added progressively at each step of the increasing iterations, and so forth. In order to achieve some of these goals, the first author [16] proposed an interpolation fractal decoding method, which is mathematically based on a new interpolation fixed-point theorem (with a constant parameter), instead of Banach's fixed-point theorem (without any control parameter).

It is well-known that the majority of image details are added at the first steps of iterations in the conventional fractal decoding, and hence the control parameter for the interpolation decoding method [16] must be set as a smaller value to achieve a better progressive decoding; however, it needs to take an extremely large number of iterations to converge. It is therefore reasonable, in terms of progressive decoding and convergence speed, to slow down the iterative process at the first stages of decoding, and to accelerate it afterwards. To achieve the goal, we proposed a new interpolation decoding scheme with variable (iteration-dependent) control parameters and proved the convergence of the decoding process mathematically. Moreover, the control parameter plays a significant role in the proposed decoding scheme and hence its determination is also an important task in this study.

2. Review of fractal image compression

In this section, we review briefly the fractal image compression and provide the general description of the technique without too many details, but those strictly necessary to understand the contents of this paper.

2.1. Fractal encoding

The fractal image compression builds on local self-similarities within real world images, in which image blocks are seen approximately as rescaled and intensity transformed copies of other blocks in the same image [17]. In fractal encoding, an input image μ_{orig} is partitioned into the nonoverlapping sub-blocks $\{\boldsymbol{R}_i\}$ (called range blocks) in a suitable manner, the union of which covers the whole image, i.e., $\cup_i \boldsymbol{R}_i = \boldsymbol{\mu}_{\mathrm{orig}}$. At the same time, a set of other sub-blocks (called domain blocks) are determined in the same image. However, the domain blocks, which are usually larger than range blocks, are allowed to be overlapping and not to cover the whole image. Typically, the range and domain blocks are square pixel blocks. The domain blocks can be obtained by sliding a window of the same size around the same image to construct the domain pool. The sliding window starts at the top left corner of the input image and moves in a given incremental step (usually larger than one pixel) in the horizontal and vertical direction, respectively. Each of the domain blocks is decimated (e.g., by pixel value averaging) to match the size of the range block, and then is compared to the input range block with an intensity affine mapping and isometric operators (rotations and flips), in the sense that the