Attractor Coding of Images

Mirek Novak
Image Coding Group, Department of Electrical Engineering,
Linköping University, 581 83 Linköping, Sweden

December 10, 1992

Abstract

This paper presents a new method for attractor coding of images. A contractive transformation, T is defined in a piecewise manner from a given image by affine mappings over triangular image regions. The image is then represented by T. Bit rates of 0.5 bits per pixel have been obtained for the 512x512 sized `LENNA` test image at 30.1 dB p-p SNR.

1 Introduction

The idea of representing images as transformations that generate them has been proposed in [1, 2, 3]. The transformations are in all these cases contractive transformations on the space of images, i.e. the transformation applied to an image produces a new image. If the transformation is contractive (see below) then recursive application of it onto an arbitrary image produces the **fixed point** (or invariant image) of the transformation. The basic problem in attractor coding methods is to calculate a transformation for a given image. Most existing methods involve a large amount of computations and searches. This problem has been treated in [4] and [5].

In this paper we will first present some of the mathematics of contractive transformations, see e.g. [6]. Next, a method to calculate transformations on images is given followed by a short note on a method for reducing the computational complexity. A determination of the amount of data required to store the transformation is made in the following section. Coding results for some test images are presented.

2 Contractive Transformations

Start with a space, $\mathcal{X}$, equipped with a metric, a distance measure $d(x, y)$, where $x, y \in \mathcal{X}$. We will only consider transformations (or mappings) of the form $w : \mathcal{X} \mapsto \mathcal{X}$.

If, for every pair of points $x, y \in \mathcal{X}$ it holds that

$$d(w(x), w(y)) \leq s \cdot d(x, y), \tag{1}$$

with the factor $s \in [0, 1[$, then w is said to be **contractive** with **contractivity factor** s. The contractivity ensures that the mapping w has a unique **fixed point** $x^* \in \mathcal{X}$ such that $w(x^*) = x^*$. Further, if we let

$$w^{(n)}(x) = \underbrace{w(w(w \ldots}_{n}(x))) = \underbrace{w \circ w \circ w \ldots}_{n}(x),$$

then for any $x \in \mathcal{X}$

$$\lim_{n \mapsto \infty} w^{(n)}(x) = x^*. \tag{2}$$

Furthermore, if we are given an arbitrary point $x_0 \in \mathcal{X}$ then

$$d(x^*, x_0) \leq \frac{1}{1-s} d(w(x_0), x_0). \tag{3}$$

This means that if we wish to find a mapping w to represent x_0 with as small error as possible, we should find a w where $d(w(x_0), x_0)$ is as small as possible and which has a small contractivity factor s.

For grayscale images the space $\mathcal{X}$ is often defined to be the space of real-valued functions defined on the supporting square of the image, or on the integer valued coordinate pairs within that square. Attractor based image coding presents us with two important problems:

- Which metric d should be chosen to measure the distance between two images?
- How should a transformation T be defined for an image f so that $d(f, T(f))$ is small?

We will make an attempt to solve these problems in the following section.

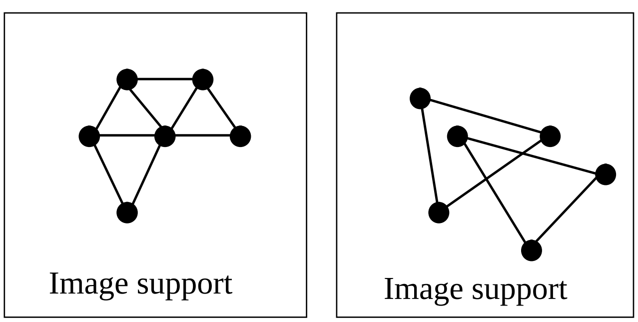
3 Image Transformations Based on Triangular Blocks

We define the space, $\mathcal{X}$, of grayscale images as the space of real-valued functions defined on the integer valued coordinates within the rectangular support of the image. The function value at position (x, y) in the image reflects the image intensity in this position.

The most common choice of metric, which will also be used here, is the root mean square **(rms)** metric. The distance $d(f, g)$ between two elements $f, g \in \mathcal{X}$ is defined as: (S is the set of (x, y) within the supporting rectangle and n(S) is the total number of points)

$$d(f,g) = \Big(\sum_{\forall(x,y)\in S} (f(x,y) - g(x,y))^2\Big)^{1/2} / n(S). \tag{4}$$

A method for calculating a transformation T that for a given image f gives a small $d(f, T(f))$ can be designed in the following way (see figure 1):



Four of the destination blocks in the image support.

Two of the source blocks in the image support.

Figure 1: Triangulation of the image support into destination and source blocks

Partition the rectangular support of the image into a set of N nonoverlapping triangular areas, called destination blocks. Each destination block is defined by a triplet of points (pairs of coordinates). Adjacent triangular regions have common corner points. Destination triangle i, $i = 1, \ldots, N$, defines a set D_i of points (x, y) in the triangle. Next, define again a set of M triangular areas in the image support, called source blocks. These can be overlapping. Each source block defines a set of points $S_j, j = 1, \ldots, M$, in the triangle.

Consider a given image, f. The idea is to define a transformation T on images in a piecewise manner, where the function values within each destination block are defined from sub-transformations of the function values within a source block.

In order to design T to give as small $d(f, T(f))$ as possible, the part of f within each destination triangle should be approximated as well as possible by a sub-transformation of f within one of the source triangles. The sub-transformations are chosen to be affine transformations. These are sufficiently complex to enable modelling of the image data, yet simple to represent.

The process of approximating the function f within one destination triangle involves the mapping of the function values in source triangle j, $f(\tilde{x}, \tilde{y})$ at the points $(\tilde{x}, \tilde{y}) \in S_j$ into new values $\hat{f}(x, y)$ at every $(x, y) \in D_i$ in destination triangle i. For this purpose we define the affine mapping from source triangle j to destination triangle i to be of the form

$$w_{ij}\left(\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ f(\tilde{x},\tilde{y}) \end{bmatrix}\right) = \begin{bmatrix} x \\ y \\ \hat{f}(x,y) \end{bmatrix} =$$

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{A_{ij}} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ f(\tilde{x},\tilde{y}) \end{bmatrix} + \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}}_{C_{ij}}.$$

The two zero entries of the matrix A_{ij} ensure that each $\hat{f}(x, y)$ is uniquely defined.

In order to calculate the parameters of A_{ij} and C_{ij} we put the following requirements on the affine mapping:

At the three corner points of the destination triangle, the mapping w_{ij} should produce function values equal to the function values of f at those points. In other words, if the corner points of the source triangle are (x_A, y_A), (x_B, y_B) and (x_C, y_C), and the corresponding points of the destination triangle are (x_α, y_α), (x_β, y_β) and (x_γ, y_γ) then it is required that

$$w_{ij}(x_A, y_A, f(x_A, y_A)) = (x_\alpha, y_\alpha, f(x_\alpha, y_\alpha)) \quad (5)$$
$$w_{ij}(x_B, y_B, f(x_B, y_B)) = (x_\beta, y_\beta, f(x_\beta, y_\beta)) \quad (6)$$
$$w_{ij}(x_C, y_C, f(x_C, y_C)) = (x_\gamma, y_\gamma, f(x_\gamma, y_\gamma)). \quad (7)$$

The equations gives us a possibility to solve 9 of the parameters in w_{ij}. The tenth parameter, a_{33} is chosen as the value that minimizes the rms distance d_i between f and $\hat{f}$ within destination triangle i. This distance is defined as

$$d_i(f,g) = \Big(\sum_{\forall(x,y)\in D_i} (f(x,y) - g(x,y))^2\Big)^{1/2}/n(D_i), \quad (8)$$

where $n(D_i)$ is the number of points in D_i. The calculation of a_{33} can be done as soon as the parameters $a_{11}, a_{12}, a_{21}, a_{22}, c_1, c_2$ are determined. These are independent of a_{33}.

If we define the following matrices:

$$X = \begin{bmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{bmatrix}. \quad (9)$$

and

$$Y = \begin{bmatrix} x_\alpha & y_\alpha & f(x_\alpha, y_\alpha) \\ x_\beta & y_\beta & f(x_\beta, y_\beta) \\ x_\gamma & y_\gamma & f(x_\gamma, y_\gamma) \end{bmatrix}. \quad (10)$$

and the column vector

$$Z = X^{-1} \begin{bmatrix} f(x_A, y_A) \\ f(x_B, y_B) \\ f(x_C, y_C) \end{bmatrix} \quad (11)$$

then we can calculate the parameters in w_{ij} from

$$R = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ c_1 & c_2 & c_3 \end{bmatrix} = X^{-1}Y - \begin{bmatrix} 0 & 0 & \vdots \\ 0 & 0 & a_{33}Z \\ 0 & 0 & \vdots \end{bmatrix} \quad (12)$$

Every combination of destination and source triangles gives a set of parameters defining a mapping w_{ij}. For each destination triangle the mapping that gives minimal distance $d_i(f, \hat{f})$ will be part of the transformation T.

If we in all w_{ij} ensure that $|a_{33}| < 1$, it can be shown that T will be contractive in some metric. This ensures that T will possess a unique fixed point.

If the distance d_i between f and $\hat{f}$ within the destination triangle is larger than a predefined value, the destination triangle is splitted in two by adding an extra point at the middle of the longest side and processing the two smaller triangles. We can be certain that the splitting process stops when the triangle degenerates into 3 points. The splitting process is usually halted at an earlier stage, accepting some areas in the image being represented with larger distortion.

After the encoding process is finished (all destination blocks are processed) we have the following definition of T: Given an arbitrary image, g, the transformed image, $T(g)$, is calculated by:

$$T(g)(x,y) = w_{ij}(\tilde{x}, \tilde{y}, g(\tilde{x}, \tilde{y})) \quad (13)$$
$$\forall i,\ (x,y) \in D_i,\ (\tilde{x}, \tilde{y}) \in S_j.$$

Due to the fact that S, the set of all points in the image is $S = \bigcup_i D_i$ (the corner points in each triangle belong to more than one D_i) we have an upper bound on the distance between f, the given image, and $T(f)$:

$$d^2(f, T(f)) \le \sum_i (n(D_i) \cdot d_i(f, \hat{f}))^2/n(S)^2. \quad (14)$$

4 Data Reduction Aspects

To represent the transformation T, each of the mappings w_{ij}, $i = 1, \ldots, N$ have to be stored. Inspecting equations (11), (10), (11) we notice that the mapping w_{ij} is defined by:

- The number of the source triangle used,
- The orientation (6 possibilities),
- The function values at the corner points of the destination triangle,
- The function values at the corners of the source triangle
- The parameter a_{33}.

It can be shown [7] that a quantization of a_{33} to 8 bits is feasible. If the total number of destination triangles is N, the number of source blocks is M

and the image has 256 levels of greyscale, then the total number, N_{tot}, of bits required to define T is

$$N_{tot} = 8(M + 4) + N({}^2\log(12M) + 16). \quad (15)$$

An 512×512 image with $M = 128$ source blocks and $N = 5707$ destination blocks yields $N_{tot} =$ 155 145 bits. This implies a bit rate of 0.59 bpp.

5 Search Reduction

In order to minimize the number of source blocks actually tested for each destination block, all source and destination blocks are categorized by a set of four numbers that are invariant under the affine mappings used. Only those source blocks whose set of invariants is close to the destination block's set of invariants are involved in the search. The invariants are defined from the moments of the greyscale distribution within each block. Moment invariants are described in [8, 7, 9]. A time reduction of 80 % can be achieved in the process of calculating the transformation T.

6 Coding Results

All images below have 256 values of greyscale. The encoding procedure for the `LENNA` image required 1500 seconds CPU on a SUN Sparcstation 2. The decoding process requires about 30 seconds. In the table below some encoding data are presented, such as the number of mappings w_{ij}, bit rate (bits per pixel) and SNR in dB (p-p).

Image	Size	# w_{ij}	bpp	SNR (dB)
LENNA	512	5707	0.59	30.1
JAS	512	3410	0.35	34.4
GIRL	256	2713	1.11	31.3

7 Discussion and Further Ideas

One major reason to use triangular subblocks of the image is that triangles are more flexible and more easily fitted to image data than square blocks. In a future implementation of the coding method an image adaptive initial triangulation scheme will be used.

Figure 2: Decoded `LENNA` image from a T with 5707 mappings.

References

[1] Michael Barnsley and Alan Sloan. A better way to compress images. *BYTE magazine*, 1988.

[2] Arnaud E. Jacquin. Image coding based on a fractal theory of iterated contractive markov operators. part ii - construction of fractal codes for digital images. *Georgia Tech Preprint 091389-017*, 1989.

[3] D.M. Monro and F. Dudbridge. Fractal approximation of image blocks. *ICASSP-92 proceedings*, 3:III – 485–488, 1992.

[4] Geir E. Øien, Skjalg Lepsøy, and Tor A. Ramstad. Reducing the complexity of a fractal-based image coder. *EUSIPCO proceedings*, 1992.

[5] Arnaud E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18–30, 1992.

[6] Michael Barnsley. *Fractals Everywhere.* Academic Press, 1988.

[7] Mirek Novak. Attractor coding of images. Licentiate thesis (to appear), Dept of EE, Linköping University, 1992.

[8] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on information theory*, pages 179–187, 1962.

[9] R. C. Gonzalez and P. Wintz. *Digital Image Processing.* Addison-Wesley, 1977.