

Non-linear fractal interpolating functions of one and two variables

R. Kobes*

Department of Physics, University of Winnipeg, Winnipeg, Manitoba, R3B 2E9 Canada

A. J. Penner†

Department of Physics, University of Manitoba, Winnipeg, Manitoba, R3T 2N2 Canada

We consider non-linear generalizations of fractal interpolating functions applied to functions of one and two variables. The use of such interpolating functions in representing images by an iterated function system is illustrated, particularly as applied to the problem of resizing images, or portions thereof.

I. INTRODUCTION

An Iterated Function System (IFS) may be used to construct fractal interpolation functions for some data [1–3]. The simplest example of interpolating a function $x(t)$, given data points (t_i, x_i) , $i = 0, 1, \dots, N$, starts with an IFS

$$W_n \begin{pmatrix} t \\ x \end{pmatrix} = \begin{pmatrix} a_n & 0 \\ c_n & 0 \end{pmatrix} \begin{pmatrix} t \\ x \end{pmatrix} + \begin{pmatrix} e_n \\ f_n \end{pmatrix}. \quad (1)$$

The coefficients a_n, c_n, e_n , and f_n determined from the conditions, for $n = 1, 2, \dots, N$,

$$W_n \begin{pmatrix} t_0 \\ x_0 \end{pmatrix} = \begin{pmatrix} t_{n-1} \\ x_{n-1} \end{pmatrix}, \quad W_n \begin{pmatrix} t_N \\ x_N \end{pmatrix} = \begin{pmatrix} t_n \\ x_n \end{pmatrix}. \quad (2)$$

which leads to

$$\begin{aligned} a_n &= \frac{t_n - t_{n-1}}{t_N - t_0}, & e_n &= \frac{t_{n-1}t_N - t_n t_0}{t_N - t_0} \\ c_n &= \frac{x_n - x_{n-1}}{t_N - t_0}, & f_n &= \frac{x_{n-1}t_N - x_n t_0}{t_N - t_0}, \end{aligned} \quad (3)$$

With this, the transformation of Eq. (1) can be written as

$$\begin{aligned} W_n(t) \equiv t' &= \frac{(t - t_0)}{(t_N - t_0)} t_n + \frac{(t - t_N)}{(t_0 - t_N)} t_{n-1} \\ W_n(x) \equiv x' &= \frac{(t' - t_{n-1})}{(t_n - t_{n-1})} x_n + \frac{(t' - t_n)}{(t_{n-1} - t_n)} x_{n-1} \end{aligned} \quad (4)$$

in which form it is apparent $W_n(x) \equiv x'$ is determined by a linear (in t) interpolating function between the points (t_{n-1}, x_{n-1}) and (t_n, x_n) . Graphs of fractal interpolating functions can then be made by applying the random iteration algorithm:

- initialize (t, x) to a point in the interval of interest

- for a set number of iterations

- randomly select a transformation $W_n(t, x)$
- plot $(t', x') = W_n(t, x)$
- set $(t, x) = (t', x')$

- end for

In this paper we consider two non-linear generalizations of such fractal interpolating functions. The first concerns how to extend the linear interpolation of Eq. (4) to higher-degree interpolations. The second generalization arises when one considers the construction of fractal interpolating functions for functions of two (or more) variables – here, even a linear interpolation of the form of Eq. (4), when applied to each variable, will result in a non-linear interpolating function. This case has an obvious application to the problem of how to represent a two-dimensional image in terms of an iterated function system; these two-dimensional interpolating functions (as a function of the pixel coordinates) can be used to represent a black-and-white image (using a Boolean function), a gray-scale image (using a scalar function), or a colour image (using a vector-valued function of the three *rgb* [red, green, blue] values). This problem has been examined extensively in the context of image compression [4–7]; in the last section we consider a related problem of using these iterated function systems to rescale images, or portions thereof.

II. FUNCTIONS OF TWO VARIABLES

We first consider a function $z(x, y)$ of two variables, and examine the problem of constructing a fractal interpolating function from the data $x_i, y_j, z_{i,j}$, where $i = 0, 1, \dots, M$, $j = 0, 1, \dots, N$, and $z_{i,j} \equiv z(x_i, y_j)$. To this end, consider the transformation

$$\begin{aligned} W_{mn}(x) &= a_{mn}x + e_{mn} \\ W_{mn}(y) &= c_{mn}y + f_{mn} \\ W_{mn}(z) &= A_{mn}x + B_{mn}y + C_{mn}xy + D_{mn} \end{aligned} \quad (5)$$

*Winnipeg Institute
for Theoretical Physics, Winnipeg, Manitoba, R3B 2E9 Canada

We then impose, for $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N$ the conditions

$$\begin{aligned} W_{mn}(z_{M,N}) &= z_{m,n}, & W_{mn}(z_{M,0}) &= z_{m,n-1}, \\ W_{mn}(z_{0,N}) &= z_{m-1,n}, & W_{mn}(z_{0,0}) &= z_{m-1,n-1} \end{aligned} \quad (6)$$

$$W_{mn} \begin{pmatrix} x_M \\ y_N \end{pmatrix} = \begin{pmatrix} x_m \\ y_n \end{pmatrix}, \quad W_{mn} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x_{m-1} \\ y_{n-1} \end{pmatrix}, \quad \text{The coefficients turn out to be}$$

$$\begin{aligned} a_{mn} &= \frac{x_m - x_{m-1}}{x_M - x_0}, & e_{mn} &= \frac{x_0 x_m - x_{m-1} x_M}{x_0 - x_M}, \\ c_{mn} &= \frac{y_n - y_{n-1}}{y_N - y_0}, & f_{mn} &= \frac{y_0 y_n - y_{n-1} y_N}{y_0 - y_N}, \\ A_{mn} &= \frac{(z_{m,n-1} - z_{m-1,n-1})y_N - (z_{m,n} - z_{m-1,n})y_0}{(x_M - x_0)(y_N - y_0)}, \\ B_{mn} &= \frac{(z_{m-1,n} - z_{m-1,n-1})x_M - (z_{m,n} - z_{m,n-1})x_0}{(x_M - x_0)(y_N - y_0)}, \\ C_{mn} &= \frac{z_{m,n} - z_{m,n-1} - z_{m-1,n} + z_{m-1,n-1}}{(x_M - x_0)(y_N - y_0)}, \\ D_{mn} &= \frac{z_{m,n}x_0y_0 - z_{m,n-1}x_0y_N - z_{m-1,n}x_Ny_0 + z_{m-1,n-1}x_Ny_N}{(x_M - x_0)(y_N - y_0)}. \end{aligned} \quad (7)$$

With these, the transformation of Eq. (5) can be written as

$$\begin{aligned} W_{mn}(x) \equiv x' &= \frac{(x - x_0)}{(x_M - x_0)} x_m + \frac{(x - x_M)}{(x_0 - x_M)} x_{m-1} \\ W_{mn}(y) \equiv y' &= \frac{(y - y_0)}{(y_N - y_0)} y_n + \frac{(y - y_N)}{(y_0 - y_N)} y_{n-1} \\ W_{mn}(z) \equiv z' &= \frac{(x' - x_{m-1})(y' - y_{n-1})}{(x_m - x_{m-1})(y_n - y_{n-1})} z_{m,n} + \frac{(x' - x_m)(y' - y_{n-1})}{(x_{m-1} - x_m)(y_n - y_{n-1})} z_{m-1,n} + \\ &+ \frac{(x' - x_{m-1})(y' - y_n)}{(x_m - x_{m-1})(y_{n-1} - y_n)} z_{m,n-1} + \frac{(x' - x_m)(y' - y_n)}{(x_{m-1} - x_m)(y_{n-1} - y_n)} z_{m-1,n-1} \end{aligned} \quad (8)$$

in which form it is apparent $W_{mn}(z) \equiv z'$ is determined by a function implementing a linear interpolation over the grid (x_{m-1}, y_{n-1}) , (x_{m-1}, y_n) , (x_m, y_{n-1}) , and (x_m, y_n) .

III. QUADRATIC INTERPOLATING FUNCTIONS

The interpolating functions considered up to now have used a linear interpolating formula between adjacent points to construct the IFS. In this section we indicate how this can be generalized to quadratic interpolations.

A. Functions of one variable

For a function $x(t)$ of one variable, using data points (t_i, x_i) , $i = 0, 1, \dots, N$, consider the transformations

$$W_n(t) = a_n t + e_n$$

$$W_n(x) = c_n t + d_n t^2 + f_n \quad (9)$$

and impose the conditions, for $n = 2, 3, \dots, N$,

$$\begin{aligned} W_n \begin{pmatrix} t_0 \\ x_0 \end{pmatrix} &= \begin{pmatrix} t_{n-2} \\ x_{n-2} \end{pmatrix}, \\ W_n \begin{pmatrix} t_M \\ x_M \end{pmatrix} &= \begin{pmatrix} t_{n-1} \\ x_{n-1} \end{pmatrix}, \\ W_n \begin{pmatrix} t_N \\ x_N \end{pmatrix} &= \begin{pmatrix} t_n \\ x_n \end{pmatrix}. \end{aligned} \quad (10)$$

The point t_M is determined as

$$t_M = \frac{(t_{n-1} - t_{n-2})}{(t_n - t_{n-2})} t_N + \frac{(t_{n-1} - t_n)}{(t_{n-2} - t_n)} t_0 \quad (11)$$

with corresponding point x_m . The coefficients of the IFS are determined as

$$\begin{aligned}
a_n &= \frac{t_n - t_{n-2}}{t_N - t_0} \\
e_n &= \frac{t_N t_{n-2} - t_0 t_n}{t_N - t_0} \\
c_n &= \frac{x_n(t_0^2 - t_m^2) + x_{n-1}(t_N^2 - t_0^2) + x_{n-2}(t_m^2 - t_N^2)}{(t_N - t_0)(t_N - t_m)(t_m - t_0)} \\
d_n &= \frac{x_n(t_m - t_0) + x_{n-1}(t_0 - t_N) + x_{n-2}(t_N - t_m)}{(t_N - t_0)(t_N - t_m)(t_m - t_0)} \\
f_n &= \frac{x_n t_m t_0(t_m - t_0) + x_{n-1} t_N t_0(t_0 - t_N) + x_{n-2} t_N t_m(t_N - t_m)}{(t_N - t_0)(t_N - t_m)(t_m - t_0)}
\end{aligned} \tag{12}$$

with which the transformation can be written as

$$\begin{aligned}
W_n(t) \equiv t' &= \frac{(t - t_0)}{(t_N - t_0)} t_n + \frac{(t - t_N)}{(t_0 - t_N)} t_{n-2} \\
W_n(x) \equiv x' &= \frac{(t' - t_{n-1})(t' - t_{n-2})}{(t_n - t_{n-1})(t_n - t_{n-2})} x_n + \frac{(t' - t_n)(t' - t_{n-2})}{(t_{n-1} - t_n)(t_{n-1} - t_{n-2})} x_{n-1} \\
&\quad + \frac{(t' - t_{n-1})(t' - t_n)}{(t_{n-2} - t_{n-1})(t_{n-2} - t_n)} x_{n-2}
\end{aligned} \tag{13}$$

In this form we see that a quadratic (in t') interpolating function is used between the points (t_n, x_n) , (t_{n-1}, x_{n-1}) , and (t_{n-2}, x_{n-2}) .

B. Functions of two variables

We next consider a function $z(x, y)$ of two variables, and construct a fractal interpolating function which employs a quadratic interpolation between points. To this end, consider the transformation

$$\begin{aligned}
W_{mn}(x) &= a_{mn}x + e_{mn} \\
W_{mn}(y) &= c_{mn}y + f_{mn} \\
W_{mn}(z) &= A_{mn}x^2y^2 + B_{mn}x^2y + C_{mn}x^2 + D_{mn}xy^2 \\
&\quad + E_{mn}xy + F_{mn}x + G_{mn}y^2 + H_{mn}y + I_{mn}
\end{aligned} \tag{14}$$

We then impose, for $m = 2, 3, \dots, M$ and $n = 2, 3, \dots, N$, the conditions

$$W_{mn} \begin{pmatrix} x_M \\ y_N \end{pmatrix} = \begin{pmatrix} x_m \\ y_n \end{pmatrix},$$

$$W_{mn} \begin{pmatrix} x_M \\ y_M \end{pmatrix} = \begin{pmatrix} x_{m-1} \\ y_{n-1} \end{pmatrix},$$

$$W_{mn} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x_{m-2} \\ y_{n-2} \end{pmatrix},$$

$$W_{mn}(z_{N,N}) = z_{m,n}, \quad W_{mn}(z_{N,m}) = z_{m,n-1},$$

$$W_{mn}(z_{N,0}) = z_{m,n-2}, \quad W_{mn}(z_{m,N}) = z_{m-1,n},$$

$$W_{mn}(z_{m,m}) = z_{m-1,n-1}, \quad W_{mn}(z_{m,0}) = z_{m-1,n-2}$$

$$W_{mn}(z_{0,N}) = z_{m-2,n}, \quad W_{mn}(z_{0,m}) = z_{m-2,n-1} \\ W_{mn}(z_{0,0}) = z_{m-2,n-2} \tag{15}$$

The points x_M and y_M are determined as

$$\begin{aligned}
x_M &= \frac{(x_{m-1} - x_{m-2})}{(x_m - x_{m-2})} x_M + \frac{(x_{m-1} - x_m)}{(x_{m-2} - x_m)} x_0 \\
y_M &= \frac{(y_{n-1} - y_{n-2})}{(y_n - y_{n-2})} y_N + \frac{(y_{n-1} - y_n)}{(y_{n-2} - y_n)} y_0,
\end{aligned} \tag{16}$$

along with the corresponding z points. The coefficients of the IFS can then be determined, by which the transformation of Eq. (15) can be written as

$$\begin{aligned}
W_{mn}(x) \equiv x' &= \frac{(x - x_0)}{(x_M - x_0)} x_m + \frac{(x - x_M)}{(x_0 - x_M)} x_{m-1} \\
W_{mn}(y) \equiv y' &= \frac{(y - y_0)}{(y_N - y_0)} y_n + \frac{(y - y_N)}{(y_0 - y_N)} y_{n-1} \\
W_{mn}(z) \equiv z' &= \frac{(x' - x_{m-2})(x' - x_{m-1})(y' - y_{n-2})(y' - y_{n-1})}{(x_m - x_{m-2})(x_m - x_{m-1})(y_n - y_{n-2})(y_n - y_{n-1})} z_{m,n} +
\end{aligned}$$

$$\begin{aligned}
& + \frac{(x' - x_{m-2})(x' - x_{m-1})(y' - y_{n-2})(y' - y_n)}{(x_m - x_{m-2})(x_m - x_{m-1})(y_{n-1} - y_{n-2})(y_{n-1} - y_n)} z_{m,n-1} + \\
& + \frac{(x' - x_{m-2})(x' - x_{m-1})(y' - y_n)(y' - y_{n-1})}{(x_m - x_{m-2})(x_m - x_{m-1})(y_{n-2} - y_n)(y_{n-2} - y_{n-1})} z_{m,n-2} + \\
& + \frac{(x' - x_m)(x' - x_{m-2})(y' - y_{n-1})(y' - y_{n-2})}{(x_{m-1} - x_m)(x_{m-1} - x_{m-2})(y_n - y_{n-1})(y_n - y_{n-2})} z_{m-1,n} + \\
& + \frac{(x' - x_m)(x' - x_{m-2})(y' - y_{n-2})(y' - y_n)}{(x_{m-1} - x_m)(x_{m-1} - x_{m-2})(y_{n-1} - y_{n-2})(y_{n-1} - y_n)} z_{m-1,n-1} + \\
& + \frac{(x' - x_m)(x' - x_{m-2})(y' - y_n)(y' - y_{n-1})}{(x_{m-1} - x_m)(x_{m-1} - x_{m-2})(y_{n-2} - y_n)(y_{n-2} - y_{n-1})} z_{m-1,n-2} + \\
& + \frac{(x' - x_m)(x' - x_{m-1})(y' - y_{n-2})(y' - y_{n-1})}{(x_{m-2} - x_m)(x_{m-2} - x_{m-1})(y_n - y_{n-2})(y_n - y_{n-1})} z_{m-2,n} + \\
& + \frac{(x' - x_m)(x' - x_{m-1})(y' - y_n)(y' - y_{n-2})}{(x_{m-2} - x_m)(x_{m-2} - x_{m-1})(y_{n-1} - y_n)(y_{n-1} - y_{n-2})} z_{m-2,n-1} + \\
& + \frac{(x' - x_m)(x' - x_{m-1})(y' - y_n)(y' - y_{n-1})}{(x_{m-2} - x_m)(x_{m-2} - x_{m-1})(y_{n-2} - y_n)(y_{n-2} - y_{n-1})} z_{m-2,n-2}
\end{aligned} \tag{17}$$

Although tedious to work out, the generalization of the preceding considerations to higher-order interpolating functions is straightforward in principle.

IV. IMAGE REPRESENTATION AND SCALING

As an application of the preceding, in this section we consider the task of representing a given colour image by an iterated function system, and then using that system to rescale a given portion of that image. This is a natural problem for an interpolating function of two variables (x, y) interpreted as pixel coordinates – the function $\vec{z}(x, y)$ in this case will be a vector-valued function having three components representing the *rgb* value of the pixel specifying the amount of red, green, and blue present.

The procedure used to scale an image of size M pixels wide by N pixels high is as follows. We first read in the *rgb* values of each pixel of the image, and use that as the data to construct a fractal interpolating function $\vec{z}(i, j)$, where $i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$ are the pixel coordinates. To then resize the image, so that the resulting image is of size $s_x M \times s_y N$, we construct a new fractal interpolating function $\vec{z}'(i, j) = \vec{z}(s_x i, s_y j)$. Applying the random iteration algorithm to $\vec{z}'(i, j)$, choosing independently a transformation index (i, j) at each stage, will then result in the rescaled image. The generalization of this procedure to rescale a portion of an image is straightforward.

As examples of the results of this procedure, consider the figures in the Appendix. We start with the image appearing in Fig. 1, and zoom in on the area of the face. The result appears in Fig. 2, together with a comparison done using a simple linear interpolation scheme. Zoom-

ing further into the area of the eye results in Fig. 3, again with a comparison of the result of a simple linear interpolation. Generally, the number of iterations needed in the random iteration algorithm to produce acceptable images is of the order of $s_x M \times s_y N$, where the original image is of size $M \times N$. Also, while slower, the quadratic fractal interpolating function typically produces, for the same number of iterations, a “smoother” looking image than the corresponding linear interpolating function. However, as with all interpolation schemes, there comes a point where such higher-order interpolating formulas actually start to produce worse results due to an artificially high sensitivity to fluctuations in the data.

Some informal tests of this procedure seems to indicate that better results are obtained for images of people, natural scenery, etc., as opposed to those containing lettering, simple geometric shapes, and similar constructs. This might be expected, given the general fractal nature of such objects in nature. However, as with all interpolating functions, it is important to remember that no structural information beyond that of the original image is being provided (for example, one could not zoom in on the face of Fig. 1 to such a degree as to see individual skin pores).

The preceding demonstrates that these non-linear fractal interpolating functions of two variables can be used in principle to represent images. The results for the rescaling of images, or portions thereof, illustrates that, at least for “natural” objects, such a representation has potential advantages. It would be interesting to extend the use of these two-dimensional functions to the case of partitioned iterated function systems, upon which much work has been done with respect to compressing images [4]. As well as in the area of image compression, such studies could lead to a more efficient representation of im-

ages by these two-dimensional functions; such partitioning involves examination of potential regions of a given image that possess a fractal nature unto themselves, as opposed to the approach used here which considers the image as a whole to have a fractal nature. Work along these directions is in progress.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

-
- [1] M. F. Barnsley, *Fractals Everywhere* (Academic Press, San Diego, CA, 1993)
 - [2] H. O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals – New Frontiers of Science* (Springer Verlag, New York, 1992).
 - [3] H. O. Peitgen and D. Saupe, *The Science of Fractal Images* (Springer Verlag, New York, 1988).
 - [4] A. Jacquin, IEEE trans. on Image Processing, January, 1992.
 - [5] M. Barnsley, L. Hurd, and L. Anson, *Fractal Image Compression* (A. K. Peters, New York, 1993).
 - [6] Y. Fisher (editor), *Fractal Image Compression: Theory and Application* (Springer Verlag, New York, 1995).
 - [7] P. R. Massopust, *Fractal Functions, Fractal Surfaces and Wavelets* (Academic Press, San Diego, CA, 1994).

Figures



FIG. 1: Original figure

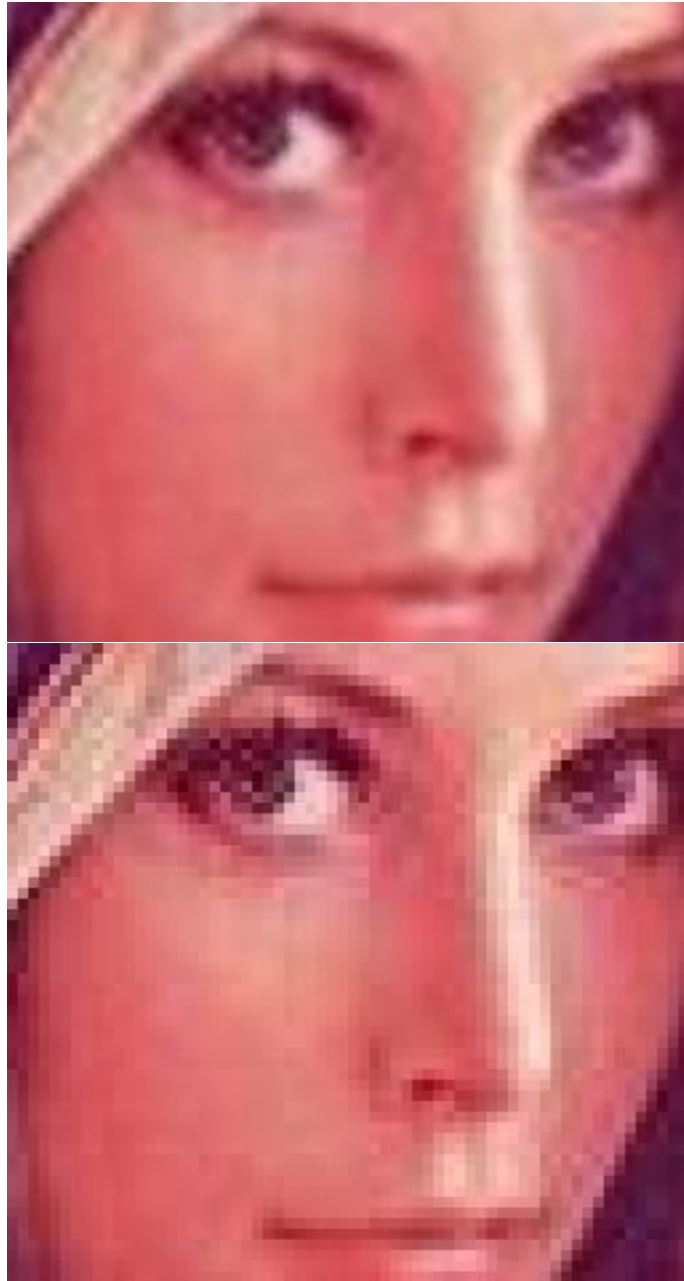


FIG. 2: Enlargement of the area around the face of Fig. 1 via a) a fractal interpolation function b) a linear interpolation

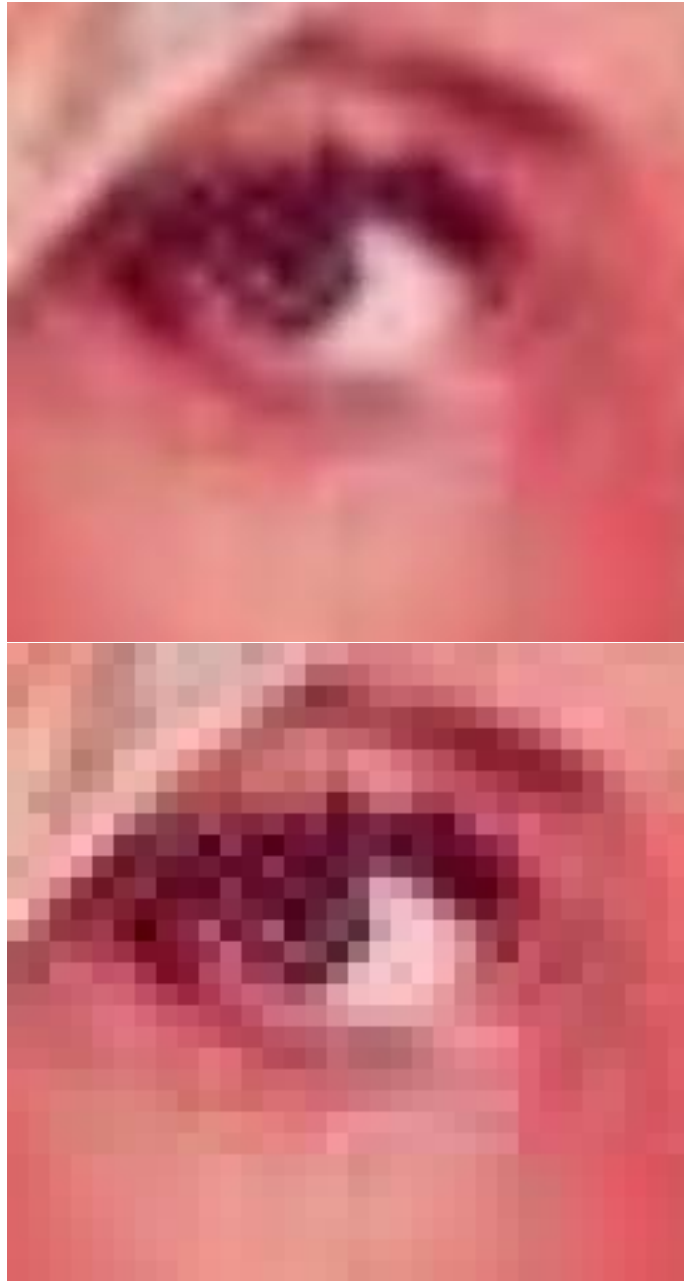


FIG. 3: Enlargement of the area around the eye of Fig. 1 via a) a fractal interpolation function b) a linear interpolation