



Original

## Exploring the use of two-dimensional piecewise-linear functions as an alternative model for representing and processing grayscale-images

Victor Manuel Jimenez-Fernandez<sup>a,\*</sup>, Héctor Vazquez-Leal<sup>a</sup>, Uriel Antonio Filobello-Nino<sup>a</sup>,  
Maribel Jimenez-Fernandez<sup>b</sup>, Luis Javier Morales-Mendoza<sup>c</sup>, Mario Gonzalez-Lee<sup>c</sup>

<sup>a</sup> Universidad Veracruzana, Facultad de Instrumentación Electrónica, Circuito Gonzalo Aguirre Beltrán s/n, Zona Universitaria, CP. 91000, Xalapa, Veracruz, Mexico

<sup>b</sup> Instituto de Ciencias Básicas, Av. Luis Castelazo Ayala s/n, Col. Industrial Ánimas, CP. 91190, Xalapa, Veracruz, Mexico

<sup>c</sup> Facultad de Ingeniería en Electrónica y Comunicaciones, Venustiano Carranza s/n, Col. Revolución, CP. 93390, Poza Rica, Veracruz, Mexico

Received 28 November 2014; accepted 14 May 2016

Available online 29 September 2016

### Abstract

Traditionally, a grayscale image is represented as a rectangular array whose internal values describe a discrete level of intensity or luminance denoted as pixel. Due to its structure and complete compatibility with matrix operators, this representation is the most widely used in image processing. Although the strong robustness of this standard is not in question, it is always enriching to have an alternative description format in order to provide not only a different image representation scheme but also an additional approach to image processing. Motivated by this fact, in this paper the viability of using continuous piecewise-linear functions of two spatial variables as an alternative model description of grayscale images is explored. Moreover, the possibility of applying this type of representation in image processing is also examined by using mapping variable transformations, here denominated as functional filters. Furthermore, it is also shown that such alternative image model can also be used in more complex tasks like tridimensional volume estimation. To verify this proposal, illustrative examples are reported, showing not only an inherent improvement in the visual perception of image, but also a practical image processing capability.

© 2016 Universidad Nacional Autónoma de México, Centro de Ciencias Aplicadas y Desarrollo Tecnológico. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

**Keywords:** Piecewise-linear; Grayscale-images; Alternative-representation

### 1. Introduction

In computer graphics literature, a graphical object is mainly defined by two fundamental characteristics: shape which refers to the geometry of the object, and attributes which is more related to intrinsic properties like color or texture (Gomes, Costa, Darsa, & Velho, 1996). Depending on the type of graphical object, one of these characteristics become more dominant, for example in digital images a description of attributes takes more relevance, while in geometric solids a shape representation is the imperative characteristic. Moreover, this difference is also

observed in the kind of mathematical model used to represent such characteristic. While discrete models are preferred to image description, functional representations are typically used in geometric and solid modeling. However, due to grayscale images present the properties of a monochrome function (Velho, Frery, & Gomes, 2009), in this paper, this type of images emerge as the more suitable kind of graphical object with the capability of joining the two fundamental graphical characteristics through a continuous functional model, preserving not only a good approximation of the object geometry but also a precise description of graphical attributes. In our proposal, the color attribute (set of luminance or pixel intensity values) is used to construct a two-dimensional function which defines the height of the graph at each coordinate in the image spatial domain. At this point, the challenge consists in obtaining a suitable continuous function from the set of discrete pixel values. In this

\* Corresponding author.

E-mail address: [vicjimenez@uv.mx](mailto:vicjimenez@uv.mx) (V.M. Jimenez-Fernandez).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

regard, it is important to clarify that the numerical technique of fitting input sparse data points by a mathematical function is not a novel strategy used in graphical objects, in fact it can be found in a variety of visualization applications like image processing (Amat, Donat, Liandrat, & Trillo, 2006; Biancardi, Lombardi, & Pacaccio, 1997; Gao, Zhang, Zhang, & Zhou, 2008; Scharinger, 1997;), computer graphics (Feng, Nishita, Jin, & Peng, 2002; Mukundan, 2012), pattern recognition (Oden, Ercil, & Buke, 2003; Subrahmonia, Cooper, & Keren, 1996; Tashk, Helfroush, & Dehghani, 2010; Tarel & Cooper, 2000), and computer-aided geometric design (Heine & Berger, 2012; Park, 2001), among others. For example, in Wu, Wang, and Chiu (2015) a spline interpolation is proposed to image up-scaling with the aim of improving resolution and sharpness, in Ho and Zeng, (2012) a bi-cubic regression is used to achieve better resolution and reduction in border zigzag effects, in Zhang, Chan, Zhang, et al. (2009) a local polynomial regression is applied to high-resolution image reconstruction, and Zhang, Chan, and Zhu (2009) presents a method for image restoration by using least-squares in polynomial regression. Nevertheless, it is important to emphasize that in all of these cases, the interpolated function is always obtained with the main purpose of processing data instead of using it as a proper representation model. In contrast, in this paper, our particular interest is to explore the viability of using continuous interpolated functions as an alternative model description of grayscale images, furthermore, its possible application to image processing is also considered. In relation to this, it must be highlighted that although there are in literature many function estimation methods based on polynomial regressions and spline approximations, our interest centers on the so-called piecewise-linear approach which in recent references (Jiménez-Fernández, Cerecedo-Nuñez, Vazquez-Leal, Beltran-Parrazal, & Filobello-Nino, 2014; Jiménez-Fernández, Vazquez-Leal, et al., 2014), has been successfully used to represent two-dimensional image curves, specifically devoted to laser projection. Moreover, in Jiménez-Fernández, Cerecedo-Nuñez, et al. (2014) a comparative study exhibited a better curve fitting performance of piecewise-linear in comparison to their spline and polynomial counterparts. Among the most emblematic

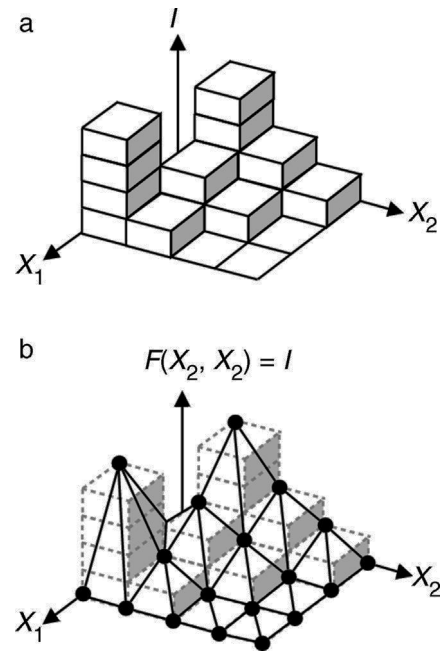


Fig. 1. Grayscale image representation. (a) Discrete pixel array. (b) Continuous piecewise-linear function.

piecewise-linear models reported in literature (Chua & Deng, 1986, 1988; Chua & Kang, 1977; Guzelis & Goknar, 1991; Kahlert & Chua, 1990; Van Bokhoven, 1998), the simplicial representation of Pedro-Julian (Julian, Desages, & Agamennoni, 1999) is taken as reference in our exploratory study. The main reasons for adopting this model are: its superior performance due to its capability to represent  $n$ -dimensional functions, and the direct programmability of the function construction methodology which is widely reported in literature (Julian, Desages, & Agamennoni, 1998; Julián, Desages, & D’Amico, 2000; Julián, 2003). The content of this paper is organized as follows. In Section 2, the basic idea of the use of continuous piecewise-linear functions to represent grayscale images is exposed. Section 3 is devoted to describe the piecewise-linear simplicial algorithm for constructing a continuous two-dimensional function. Examples

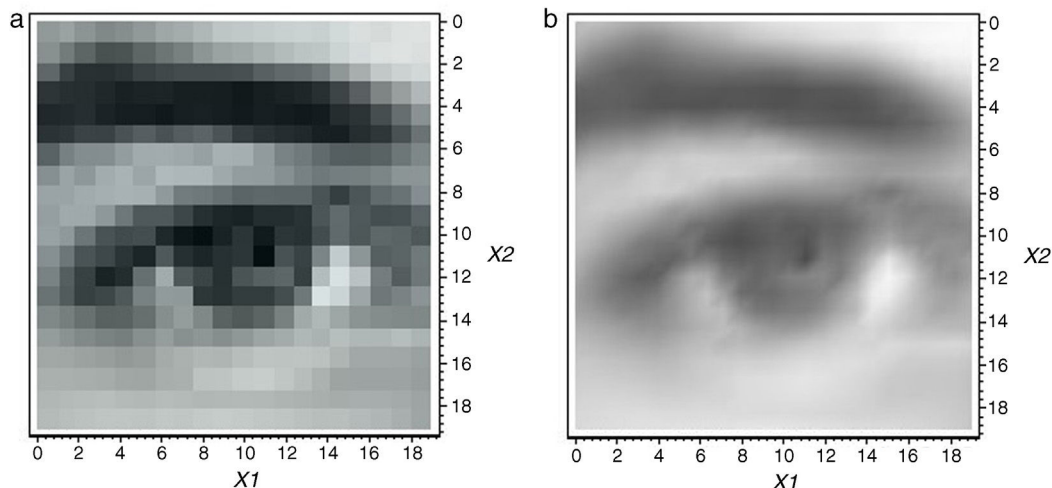


Fig. 2. (20 × 20)-resolution grayscale image. (a) In a pixel format. (b) In a piecewise-linear functional format.

and discussions about the experiment results of the proposed approach are given in Section 4. Finally, conclusions are drawn in Section 5.

### 2. Basic idea

Let it consider a grayscale image  $I$  constituted by  $(n \times m)$  representative components of brightness-level,  $I_{i,j}$ , for  $i = \{1, 2, \dots, m\}$ , and  $j = \{1, 2, \dots, n\}$ .

In a traditional image representation model, these components are picture elements (pixels) that can be considered as sampling values defined in a rectangular spatial grid data  $X_1 X_2$ . Under such approach, an image perception depends on how each point in space associates a color level to that point. However, if the pixel sampling values are linear interpolated around each  $(i, j)$ -th coordinate with the aim to construct a continuous piecewise-linear function  $F(X_1, X_2) = I$ , then the resulting  $F(X) = I$  can be used as an alternative representation model. Basically, this model maps the rectangular set of gray levels into a continuous function whose geometric graph depicts the image itself, namely, it means a mathematical recasting from a discrete model to a continuous one. In order to clarify terms, this concept is graphically illustrated in Fig. 1.

Under this alternative representation, a better performance can be obtained in the way the image is visually perceived. Although a more detailed analysis of this point is given in Section 4, it is relevant here to notice, the immediate improvement that can be achieved in resolution and visual acuity. It means, compared with the traditional array representation, an image enhancement is directly obtained from the continuous functional model due to additional sub-graylevels are incorporated into the function.

This effect is more evident in low resolution images. To illustrate this, it should be consider the examples of the visual comparisons shown in Fig. 2 and Fig. 3, respectively.

### 3. Function construction algorithm

Based on the simplicial piecewise-linear methodology of (Julian, 1999), our proposal for the construction of the continuous function  $F(X) = I$  is summarized as follows.

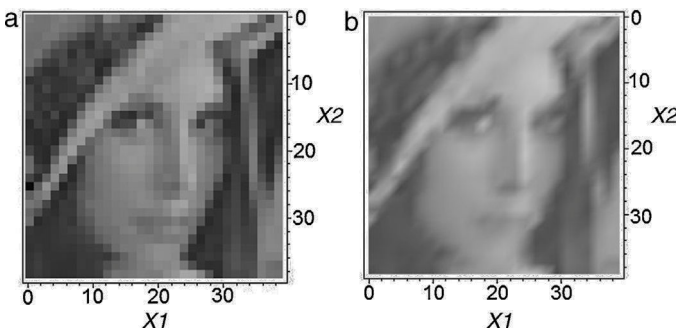


Fig. 3. (40 × 40)-resolution grayscale image. (a) In a pixel format. (b) In a piecewise-linear functional format.

Function construction algorithm.	
<b>INPUT:</b>	$I_{i,j}$ -Grayscale image with $(m \ n)$ -pixel resolution, $i = \{1,2,\dots,m\}, j = \{1,2,\dots,n\}$
<b>OUTPUT:</b>	$F(X_1, X_2) = I$ -Continuous piecewise linear function
for $i$ from 1 to $m$ do	-Image reading
for $j$ from 1 to $n$ do	
$(X_1, X_2, F_{i,j}) \leftarrow (i, j, I_{i,j})$	
$D = \{(0, 0, F_{1,1}), \dots, (X_1, X_2, F_{i,j}), \dots, (m-1, n-1, F_{m,n})\}$	
end	
end	
	-Simplicial partition
for $k = \{1, 2, \dots, (m-1)\}$ do	$P_h(X_1, X_2) = \{X_1 = k\}$
for $k = \{1, 2, \dots, (n-1)\}$ do	$P_v(X_1, X_2) = \{X_2 = k\}$
for $k = \{(m+1), (m+2), \dots, 0, 1, \dots, (n-2), (n-1)\}$ do	$P_t(X_1, X_2) = \{X_1 - X_2 = k\}$
	-Vertices sorting
$\{class_0 \text{ vertex}\} V_0 \leftarrow \{(0,0)\}$	
$\{class_1 \text{ vertex}\} V_1 \leftarrow \{(i,0), \dots, (0,j)\}$ ,	
$i = \{1, 2, \dots, m\}, j = \{1, 2, \dots, n\}$	
$\{class_2 \text{ vertex}\} V_2 \leftarrow \{(i,j) \mid i = \{2, \dots, m\}$	
for each $j = \{2, \dots, n\}$	
$V = [V_1 : V_2 : V_3]$	
	-Cartesian Product of partition equations
$P_S = P_h \ P_v = \{(P_{X_1}^{(a)}, P_{X_2}^{(b)})\}$	
where $(a,b) \in \{1, \dots, (m-1)\} \times \{1, \dots, (n-1)\}$	
	-Basis function
$(f_a, f_b) = \frac{1}{4} \{ \ f_a + f_b\ , \ f_a - f_b\ , \ f_a\  + \ f_b\ , \ f_a\  - \ f_b\  \}$	
	-Matrix domain function
$\Lambda(X_1, X_2) = [l^0, l^1, \dots, l^l]$ with $l^i(X_1, X_2) = l^i(P_{X_1}^{(a)}, P_{X_2}^{(b)})$	
and $l = \{1, 2, \dots, (m-1)(n-1)\}$	
$\Lambda_V = [\Lambda^l(V)]^T$ $(m \ n)$ -matrix	
	-Codomain function
$\mathbf{B} = [F_{0,0} : F_{1,0}, \dots, F_{m,0} : F_{0,1}, \dots, F_{0,n} : F_{1,1}, \dots, F_{i,j}, \dots, F_{m,n}]^T$	
$F(X_1, X_2) = \Lambda_V^{-1} \mathbf{B}$	

This algorithm is divided into seven consecutive numerical steps: Image reading, Simplicial partition, Vertices sorting, Symbolic basis-function generation, Basis function evaluation, and Codomain and interpolated function construction.

#### 3.1. Image reading

In this step, every image value  $I_{i,j}$  is read from the pixel matrix array, and then it is collected and associated to its matrix position in order to form the data structure

$$D = \{(0, 0, F_{1,1}), \dots, (X_1, X_2, F_{i,j}), \dots, (m-1, n-1, F_{m,n})\}.$$

#### 3.2. Simplicial partition

The  $(i, j)$ -th matrix position is mapped to its corresponding spatial coordinate in a function domain  $X_1 X_2$ . From a geometric

perspective, such function domain is considered as a linear  $\{X_1X_2\}$ -plane; moreover, the discrete spatial coordinates can be seen as the breakpoints of an arbitrary continuous piecewise-linear function  $F(X_1, X_2)$ . In accordance with the interpolation methodology of Pedro-Julian, the function domain must be equally sized partitioned into triangular blocks denoted simplices. The simplicial partition of the domain is done by line equations which are directed to three types of traces: horizontal ( $P_h(X_1, X_2)=X_1=k$ ), vertical ( $P_v(X_1, X_2)=X_2=k$ ), and transversal ( $P_t(X_1, X_2)=X_1-X_2=k$ ). Particularly, for each of these traces the sweep of  $k$  must be done in accordance with the boundaries existing along to the  $X_1$  and  $X_2$  axis. For example, if the restrictions  $X_1=m$  and  $X_2=m$  are given, then three  $k$ -sweeps:  $k=1, 2, \dots, (m-1)$ ,  $k=1, 2, \dots, (n-1)$ , and  $k=(-m+1), (-m+2), \dots, 0, 1, \dots, (n-2), (n-1)$  will correspond to the vertical, horizontal, and transversal traces, respectively.

### 3.3. Vertices sorting

The vertices (piecewise-linear function breakpoints) must be ordered according to their class in the vector  $V=[V_1 : V_2 : V_3]$ . For the domain  $X_1X_2$ , the class-zero vertex ( $V_0$ ) is the point (0, 0) which is located at the origin of the coordinate system, the class-one vertices ( $V_1$ ) are the coordinates  $(i, 0)$  and  $(0, j)$  (for  $i=\{1, 2, \dots, m\}$  and  $j=\{1, 2, \dots, n\}$ , respectively), and the set of class-two vertices ( $V_2$ ) is constituted by the coordinates  $(i, j)$  with the  $j$ -th index running as  $j=1, 2, \dots, m$ , for each  $i$ -th value from 1 to  $n$ .

### 3.4. Symbolic basis-functions generation

In the simplicial partition of  $X_1X_2$ -domain will be a set of  $(m-1)$  horizontal line equations ( $P_h(X_1X_2)$ ), and a set of  $(n-1)$  vertical line equations ( $P_v(X_1X_2)$ ). If a cartesian product  $P_S = P_h \times P_v$  between these two sets is formed, then each ordered pair (whose first component is an equation member of  $P_h$  and whose second component is an equation member of  $P_v$ ) is evaluated to form the basis-function vector

$$\mathbf{\Lambda}^l(X_1, X_2) = [\Lambda^0, \Lambda^1, \dots, \Lambda^l] \quad \text{with} \quad \Lambda(X_1, X_2) = \gamma^l(P_{X_1}^{(a)}, P_{X_2}^{(b)}), \text{ for } l=\{1, 2, \dots, (m-1)(n-1)\}, \text{ and } \gamma(\cdot) \text{ defined as } \gamma(f_a, f_b) = \frac{1}{4}\{|-f_a| + f_b| - |-f_a + f_b| + |-f_a| + |f_b| - |-f_a + f_b|\}$$

### 3.5. Basis-functions evaluation

The vector  $\mathbf{\Lambda}$  is evaluated at each ordered vertex. It is important to observe the similarity that exists between the methodology that is used to order vertices and that used to construct the matrix  $\mathbf{\Lambda}_V$  which is constituted by the rows of evaluated vectors  $\mathbf{\Lambda}^l(V)$ . That is

$$\mathbf{\Lambda}_V = \begin{bmatrix} \Lambda^0(V_0) & \Lambda^1(V_0) & \dots & \Lambda^l(V_0) \\ \Lambda^0(V_1) & \Lambda^1(V_1) & \dots & \Lambda^l(V_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda^r(V_r) & \Lambda^r(V_r) & \dots & \Lambda^r(V_r) \end{bmatrix} \tag{1}$$

where  $r$  indicates the number of vertices denoted as  $r=(m-1) \times (n-1)$ .

### 3.6. Codomain and interpolated function construction

Two operations are involved in this step, in the first one the values of function at the breakpoint coordinates are provided by following a sorted order associated with the vertices  $\{V_0, V_1, \dots, V_r\}$ . These values are collected in vector  $\mathbf{B}$  and written as follows

$$\mathbf{B} = [B_0 B_1 \dots B_r]^T \tag{2}$$

Finally, in the second one, the mathematical operation  $F(X_1, X_2) = \mathbf{\Lambda}_V^{-1} \mathbf{B}$  is performed.

#### 3.6.1. Example of algorithm output:

In order to illustrate the output of the function construction algorithm here described, consider the  $(4 \times 4)$ -pixel image of Fig. 4.

After applying all the algorithm steps, the following equation is obtained:

$$\begin{aligned} F(X_1, X_2) = & 1 - 3|X_1 - 1| - |X_1| + 0.25|X_1 - |X_1|| - 0.25||X_1| + X_1| + 0.5|X_2| + \\ & 0.25||X_2| + X_2| + 63.5||X_1| + X_2| + 0.75||X_1 - 1| + X_1 - 1| - 0.75|X_1 - 2|X_1 - 1|| - \\ & 32.75|X_1 - 2| - 1||X_1 - 2| + X_1 - 2| + |X_1 - 3|X_1 - 2|| - 0.25|X_2 - 1|X_2|| - 33.25|X_2 - 1| - \\ & 0.5||X_2 - 1| + X_2 - 1| - 30|X_2 - 2| + 0.75||X_2 - 2| + X_2 - 2| + 0.5|X_2 - 2|X_2 - 1|| - 0.75|X_2 - \\ & 3|X_2 - 2|| - 63.5|X_1 - 1|X_2|| - 31||X_1| + X_2 - 1| - 63.25|X_1 - X_2| + 31|X_1 - 1|X_2 - 1|| + \\ & 30.75|X_1 - X_2 + 1| - 33||X_1| + X_2 - 2| + 33|X_1 - 1|X_2 - 2|| + 33|X_1 - X_2 + 2| - 31.25||X_1 - 1| + \\ & X_2| + 31.25|X_1 - 2|X_2|| + 31|X_1 - X_2 - 1| - 1.5||X_1 - 1| + X_2 - 1| + 1.5|X_1 - 2|X_2 - 1|| + \\ & 0.25||X_1 - 1| + X_2 - 2| - 0.25|X_1 - 2|X_2 - 2|| - 32.25||X_1 - 2| + X_2| + 32.25|X_1 - 3|X_2|| + \\ & 32.25|X_1 - X_2 - 2| - 0.25|X_1 - 3|X_2 - 1|| + 0.25||X_1 - 2| + X_2 - 1| + 1.25||X_1 - 2| + \\ & X_2 - 2| - 1.25|X_1 - 3|X_2 - 2|| \end{aligned}$$

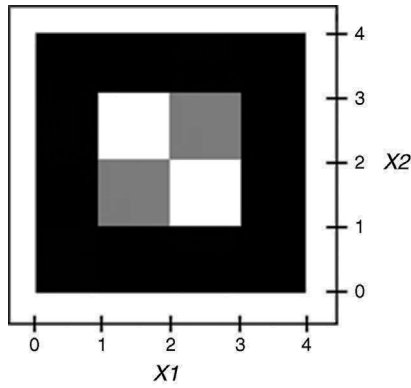


Fig. 4. (4 × 4) grayscale image in a traditional pixel format.

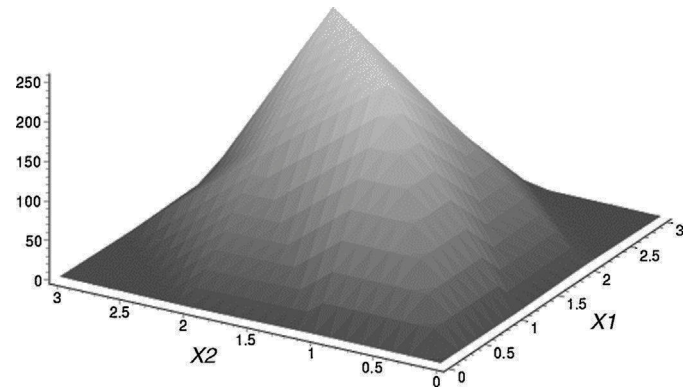


Fig. 5. Piecewise-linear function for the image of Fig. 4.

A plot of this function is show in Fig. 5.

#### 4. Experiments and discussion

This section explores, by means of illustrative examples, the processing capability of the alternative image representation model. Specifically, two characteristics are examined: the sharpness that is itself present in the image as a result of being described by a continuous function, and the contrast that is obtained by applying functional filters. Additional to these experiments, the potential usage of the model in image analysis techniques such as volume computation is also discussed.

##### 4.1. Sharpness

In regards to sharpness, although the level of clarity of details in images is usually analyzed by considering two fundamental factors: the edge contrast defined as acutance, and the amount of information within the image denoted as resolution, in our comparative discussion only acutance is approached, this because resolution is a factor more closely related to the number of pixels per spatial dimensions. This fact, definitely restricts its application to only pixel image based representations. As reference, let it consider the pixel and function interpolated images shown in Fig. 6.

Taking into account that acutance describes how quickly image information transitions at an edge, from the above figure, it can be seen that while in the pixel format (Fig. 6(a)) a remarkable edge transition can be noted, in the function interpolated format (Fig. 6(b)) a smoothing edge transition is observed. From this observation, it is possible to conclude that in Fig. 6(b) a lower acutance is present. In this sense, it is important to notice that although it could be considered preferable to have a high acutance (because it let to obtain sharper and more detailed borders), in low resolution images it is not the best option. This effect is more notorious in three image details of Fig. 6: in the hat, in the cheek, and in the shoulder, where it can be seen that besides to look like less noisy, Fig. 6(b) also shows smoothing traces that contrast with the squared or zigzag traces of Fig. 6(a).

##### 4.2. Contrast

Contrast defines the level of separation between the darkest and brightest areas. A high contrast image exhibits a full range of tones, from black to white, with dark shadows and bright highlights. A low contrast image, exhibits a poor difference between its lights and darks areas what results in a flat appearance. In order to illustrates the viability of modifying contrast in images described by a function interpolated representation, Fig. 7 shows the results obtained in the reference image after applying an exponential functional filter. At this point it is important to clarify that the concept of functional filter has been introduced to describe a mathematical operation that is applied to the continuous interpolated image representation. It has been denoted here as functional to remark the difference with the common mask filters used the traditional pixel representation format of images.

In order to simplify nomenclature, let it be the notation  $F(X_1, X_2) = fpwl$ , where  $F(X_1, X_2)$  is the piecewise-linear function that represent the reference image depicted in Fig. 6. In the experiment of contrast variation, the functional filter whose result and characteristic curve are shown in Fig. 7, is given by the mapping formulation

$$F_c(X_1, X_2) = \frac{255}{1 + e^{-\frac{1}{2}(fpwl-20)}} \quad (3)$$

where  $F_c(X_1, X_2)$  is used to denote the continuous function which represents the modified contrast image of Fig. 7(a).

Additional experiments results of contrast variation are provided in Fig. 8 and Fig. 9, where the corresponding functional



Fig. 6. (100 × 100) grayscale image: (a) in a traditional pixel format and (b) in a piecewise-linear function interpolated format.

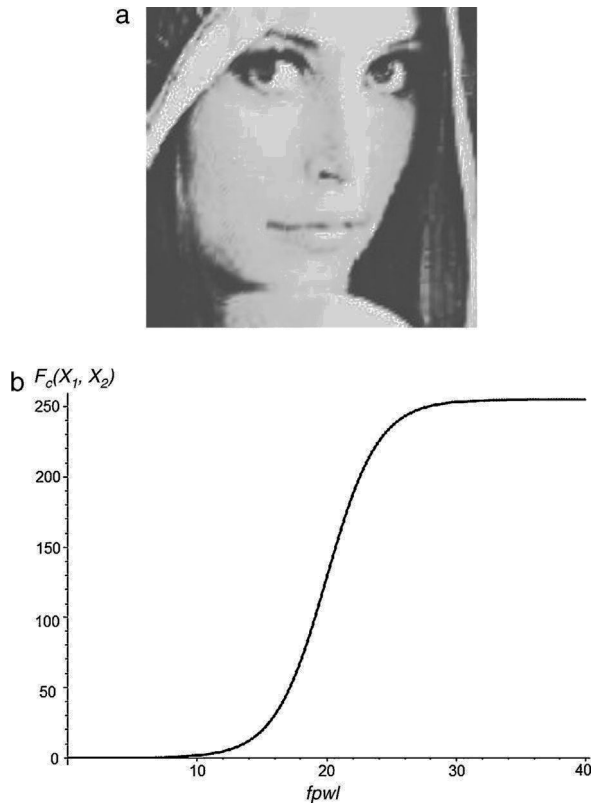


Fig. 7. Results obtained from contrast experiments in the  $(100 \times 100)$  grayscale interpolated function image of reference: (a) image with modified contrast and (b) characteristic curve of the functional filter.

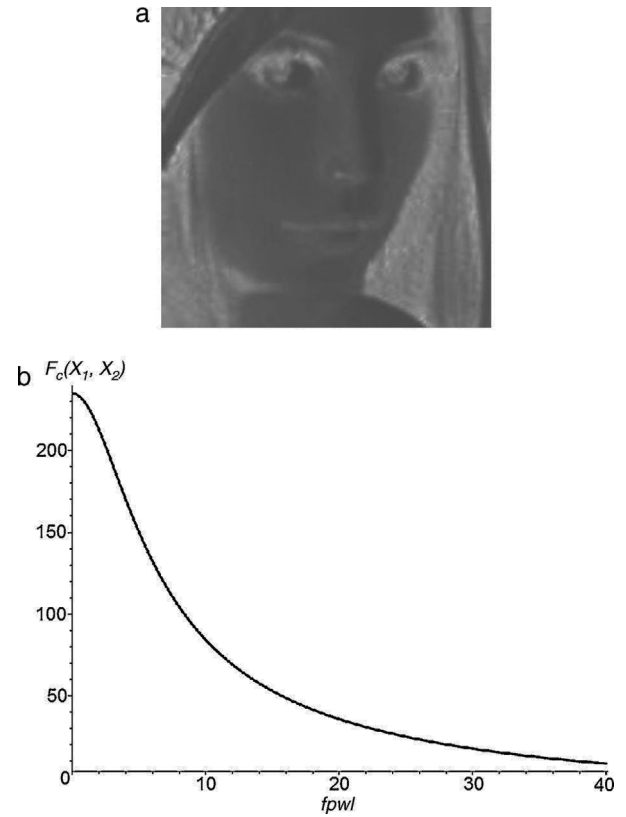


Fig. 9. More results obtained from contrast experiments in the  $(100 \times 100)$  grayscale image function of reference: (a) image with modified contrast and (b) characteristic curve of the functional filter (5).

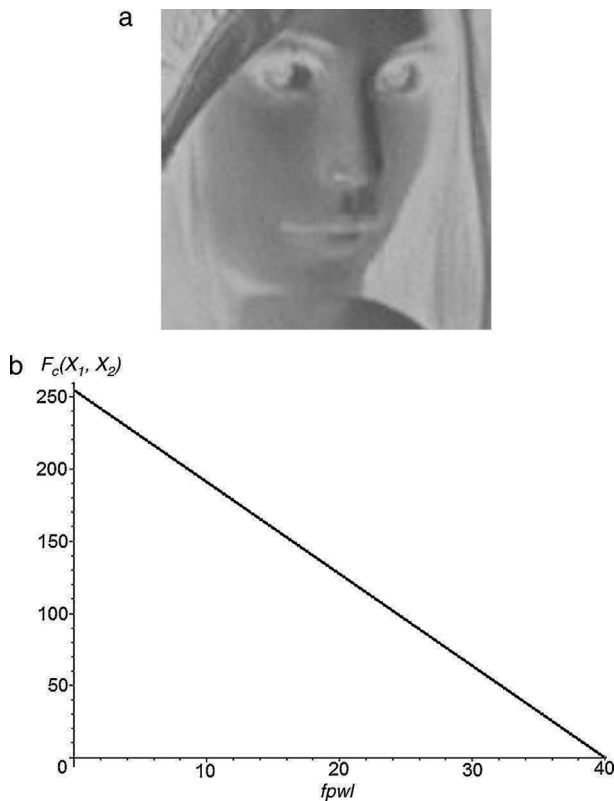


Fig. 8. Additional results obtained from contrast experiments in the  $(100 \times 100)$  grayscale interpolated function image of reference: (a) image with modified contrast and (b) characteristic curve of the functional filter.

filters used to obtain such visual effects are Eq. (4) and Eq. (5), respectively.

$$F_c(X_1, X_2) = \frac{255}{40} fpwl + 255 \tag{4}$$

$$F_c(X_1, X_2) = \frac{255}{\sqrt{1 + 0.05 fpwl^2}} \tag{5}$$

### 4.3. Volume estimation

The last experiment, here reported, is related to volume estimation by image processing. In that regard, it is important to mention that dealing with an image representation based on a continuous function exhibits a direct advantage in comparison with the conventional pixel model. On the one hand, it must be observed that due to  $F(X_1, X_2) = I$  is an explicit expression, then from a purely mathematical point of view, the approximated volume estimation can be computed as

$$V = \delta \int_{X_1^{(i)}}^{X_1^{(f)}} \int_{X_2^{(i)}}^{X_2^{(f)}} F(X_1, X_2) dX_1 dX_2 \tag{6}$$

where the integrand ranges:  $[X_1^{(i)}, X_1^{(f)}]$  and  $[X_2^{(i)}, X_2^{(f)}]$ , define the selected surface (function domain), and  $\delta$  corresponds to the scale factor which must be included in order to denormalize the computed volume results. On the other hand, if a traditional pixel strategy is used to volume estimation, firstly, a

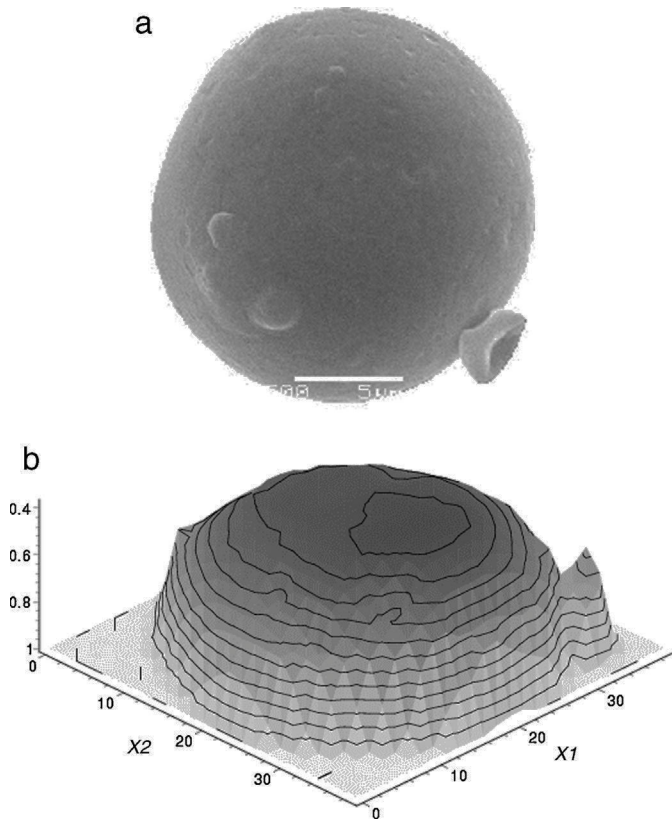


Fig. 10. Microsphere image of reference: (a) original grayscale image and (b) top geometry view of the scaled image to  $(40 \times 40)$  resolution.

tridimensional object compound of two-dimensional contours acquired from segments slices must be constructed, and secondly, the approximated volume must be calculated by adding each consecutive slice surface. From a numerical point of view, under the traditional discrete strategy, the volume computation results from a summation procedure, where every three-dimensional pixel (usually named as voxel) is added on a regular grid space. In order to explore the potential application of the alternative image representation in volume estimation, the microsphere of Fig. 10 is considered.

In our experiment, the algorithm stated in Section 3 was implemented in Maple Software version 15 to construct the piecewise-linear function  $F(X_1, X_2)=I$  of the  $(40 \times 40)$  grayscale image of Fig. 10(b). After that, Eq. (6) was used to compute its volume, resulting a normalized ( $\delta=10$ ) value of 276.141761 cubic units, when the numerical method of Monte-Carlo is applied with a tolerance constraint of  $0.5 \times 10^{-14}$ .

## 5. Conclusion

In this paper the viability of using a continuous two-dimensional piecewise-linear function to describe an arbitrary grayscale image was explored. The function construction methodology of Pedro-Julian was used as reference model to fit the set of discrete graylevel values in a simplicial piecewise-linear geometric surface. The plot of the resulting function showed the possibility of sketching not only the image shape but also incorporating additional attributes like

gray level color. Experiments in low-resolution images showed that piecewise-linear functions are a valid alternative model for image representation. Besides of the evident improvement in the visual perception that results from mapping a discrete pixel scheme into a continuous plot, a smoothing transition can also be observed in consecutive high-contrast gray levels. This natural property let it perceive a better resolution and a reduction in border zigzag effects. Finally, it must be highlighted that the alternative representation scheme also exhibited compatibility with image processing operations, particularly, successfully results were obtained by applying functional filters for image-contrast as well as for computing three-dimensional volume estimation.

## Conflict of interest

The authors have no conflicts of interest to declare.

## References

- Amat, S., Donat, R., Liandrat, J., & Trillo, J. C. (2006). Analysis of a new non-linear subdivision scheme. *Applications in image processing. Foundations of Computational Mathematics*, 6(2), 193–225.
- Biancardi, A., Lombardi, L., & Pacaccio, V. (1997). Improvements to image magnification. *International Conference on Image Analysis and Processing*, 1(1), 142–149.
- Chua, L., & Deng, A. C. (1986). Canonical piecewise-linear modeling. *IEEE Transactions on Circuits and Systems*, 33(5), 511–525.
- Chua, L. O., & Deng, A. C. (1988). Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, 35(1), 101–111.
- Chua, L. O., & Kang, S. M. (1977). Section-wise piecewise-linear functions: Canonical representation, properties, and applications. *Proceedings of the IEEE*, 65(6), 915–929.
- Feng, J., Nishita, T., Jin, X., & Peng, Q. (2002). B-spline free-form deformation of polygonal objects as trimmed Bezier surfaces. *The Visual Computer*, 18(8), 493–510.
- Gao, S., Zhang, C., Zhang, Y., & Zhou, Y. (2008). Medical image zooming algorithm based on bivariate rational interpolation. *International Symposium on Visual Computing*, (5359), 672–681.
- Gomes, J., Costa, B., Darsa, L., & Velho, L. (1996). Graphical objects. *The Visual Computer*, 12(6), 269–282.
- Guzelis, C., & Goknar, I. C. (1991). A canonical representation for piecewise-affine maps and its applications to circuit analysis. *IEEE Transactions on Circuits and Systems*, 38(11), 1342–1354.
- Heine, A., & Berger, M. (2012). MOCAD: A tool for graphical and interactive calculation and optimization of cam mechanisms and motion control systems. In *Advances in mechanisms design*. pp. 129–135. Netherlands: Springer.
- Ho, T. C., & Zeng, B. (2012). Image super-resolution by curve fitting in the threshold decomposition domain. *Journal of Visual Communication and Image Representation*, 23(1), 208–221.
- Jiménez-Fernández, V. M., Cerecedo-Núñez, H. H., Vazquez-Leal, H., Beltran-Parrazal, L., & Filobello-Nino, U. (2014). A parametric piecewise-linear approach to laser projection. *Computational and Applied Mathematics*, 33(3), 841–858.
- Jiménez-Fernández, V. M., Vazquez-Leal, H., Filobello-Nino, U., Cerecedo-Núñez, H. H., Padilla-Sosa, P., & Beltrán-Parrazal, L. (2014). A comparative study between piecewise-linear and point-based methodologies for galvanometer mirror systems. *Revista Facultad de Ingeniería Universidad de Antioquia*, 73, 124–133.
- Julian, P. (1999). *A high level canonical piecewise linear representation: Theory and applications* Doctoral dissertation, Ph.D. Thesis. pp. 52–89. Bahía Blanca, Argentina: Universidad Nacional del Sur.

- Julián, P. M. (2003). The complete canonical piecewise-linear representation: Functional form for minimal degenerate intersections. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 50(3), 387–396.
- Julian, P., Desages, A., & Agamennoni, O. (1998). On the high level canonical representation of piecewise linear functions. In *ISCAS'98. Proceedings of the 1998 IEEE international symposium on circuits and systems*, Vol. 6 (pp. 314–317).
- Julian, P., Desages, A., & Agamennoni, O. (1999). High-level canonical piecewise linear representation using a simplicial partition. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 46(4), 463–480.
- Julián, P., Desages, A., & D'Amico, B. (2000). Orthonormal high-level canonical PWL functions with applications to model reduction. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(5), 702–712.
- Kahlert, C., & Chua, L. O. (1990). A generalized canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems I*, 37(3), 373–383.
- Mukundan, R. (2012). *Advanced methods in computer graphics, with examples in OpenGL*. pp. 1–312. London: Springer-Verlag.
- Oden, C., Ercil, A., & Buke, B. (2003). Combining implicit polynomials and geometric features for hand recognition. *Pattern Recognition Letters*, 24(13), 2145–2152.
- Park, H. (2001). An approximate lofting approach for B-spline surface fitting to functional surfaces. *The International Journal of Advanced Manufacturing Technology*, 18(7), 474–482.
- Scharinger, J. (1997). Image compression by multilevel polynomial interpolation and wavelet texture coding. In *International conference on computer aided systems theory EUROCAST-97*, Vol. 1333 (pp. 429–443).
- Subrahmonia, J., Cooper, D. B., & Keren, D. (1996). Practical reliable Bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5), 505–519.
- Tashk, A., Helfroush, M. S., & Dehghani, M. J. (2010). A Chebyshev/Legendre polynomial interpolation approach for fingerprint orientation estimation smoothing and prediction. *Journal of Zhejiang University Science C*, 11(12), 976–988.
- Tarel, J. P., & Cooper, D. B. (2000). The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), 663–674.
- van Bokhoven, W. M. G. (1998). *Piecewise linear modelling and analysis*, Norwell, MA, USA. pp. 31–50.
- Velho, L., Frery, A. C., & Gomes, J. (2009). *Image processing for computer graphics and vision*. pp. 133–134. London, England: Texts in Computer Science.
- Wu, W. C., Wang, T. H., & Chiu, C. T. (2015). Edge curve scaling and smoothing with cubic spline interpolation for image up-scaling. *Journal of Signal Processing Systems*, 78(1), 95–113.
- Zhang, Z. G., Chan, S. C., Zhang, X., Lam, E. Y., Wu, E. X., & Hu, Y. (2009). High-resolution reconstruction of human brain MRI image based on local polynomial regression. In *2009 4th international IEEE/EMBS conference on neural engineering* (pp. 245–248).
- Zhang, Z. G., Chan, S. C., & Zhu, Z. Y. (2009). A new two-stage method for restoration of images corrupted by Gaussian and impulse noises using local polynomial regression and edge preserving regularization. In *2009 IEEE international symposium on circuits and systems* (pp. 948–951).