Kolmogorov Superposition Theorem and Wavelets for image compression

Pierre-Emmanuel Leni, Yohan D. Fougerolle, and Frédéric Truchetet

Université de Bourgogne, Laboratoire LE2I, UMR CNRS 5158,
12 rue de la fonderie, 71200 Le Creusot, France

ABSTRACT

We propose a new compression approached based on the decomposition of images into continuous monovariate functions, which provide adaptability over the quantity of information taken into account to define the monovariate functions: only a fraction of the pixels of the original image have to be contained in the network used to build the correspondence between monovariate functions. The Kolmogorov Superposition Theorem (KST) stands that any multivariate functions can be decomposed into sums and compositions of monovariate functions. The implementation of the decomposition proposed by Igelnik, and modified for image processing, is combined with a wavelet decomposition, where the low frequencies will be represented with the highest accuracy, and the high frequencies representation will benefit from the adaptive aspect of our method to achieve image compression.

Our main contribution is the proposition of a new compression scheme, in which we combine KST and multiresolution approach. Taking advantage of the KST decomposition scheme, we use a decomposition into simplified monovariate functions to compress the high frequencies. We detail our approach and the different methods used to simplify the monovariate functions. We present the reconstruction quality as a function of the quantity of pixels contained in monovariate functions, as well as the image reconstructions obtained with each simplification approach.

1. INTRODUCTION

The Superposition Theorem is the solution of one of the 23 mathematical problems conjectured by Hilbert in 1900. Kolmogorov has proved that continuous multivariate functions can be expressed as sums and compositions of monovariate functions. The KST, reformulated and simplified by Sprecher in Ref. 1, 2, can be written as:

THEOREM 1.1 (KOLMOGOROV SUPERPOSITION THEOREM). *Every continuous function defined on the identity hypercube, $f : [0,1]^d \longrightarrow \mathbb{R}$, can be written as sums and compositions of continuous monovariate functions as:*

$$f(x_1, ..., x_d) = \sum_{n=0}^{2d} g_n\Big(\sum_{i=1}^{d} \lambda_i \psi(x_i + b_n)\Big), \tag{1}$$

with ψ continuous function, λ_i and b constants. ψ is called inner function and g external function. Coordinates $x_i, i \in [\![1, d]\!]$ of each dimension are combined into a real number by a hash function (obtained by linear combinations of inner functions ψ) that is associated to corresponding value of f for these coordinates by the external function g.

Igelnik has presented in Ref. 3 an approximating construction that provides flexibility and modification perspectives over the monovariate function construction. Using Igelnik's approximation network, the image can be represented as a superposition of layers, *i.e.* a superposition of images with a fixed resolution. The constructed network can be reduced to a fraction of the pixels of the whole image: the smaller the tiles, the larger the quantity of information. We study the reconstruction quality using monovariate functions containing only a fraction of the original image pixels. To improve the reconstruction quality, we apply this decomposition on images of details

Further author information:
pierre-emmanuel.leni@u-bourgogne.fr
yohan.fougerolle@u-bourgogne.fr
frederic.truchetet@u-bourgogne.fr

obtained by a wavelet decomposition: external functions obtained from the decomposition of images of details can be simplified, thus decreasing the quantity of pixels required to reconstruct the original image. Two methods can be considered to simplify the external functions. We detail both, and present the reconstructions obtained using each simplification, as well as the simplification performance as a function of the reconstruction accuracy.

The structure of the paper is as follows: we present the decomposition algorithm in section 2. In section 3, we present the KST decomposition combined wavelets and the methods implemented to simplify the monovariate functions and compress the image. In the last section, we present our conclusions and several promising research perspectives.

Our contributions include the characterization of a new compression approach. We combine KST decomposition with wavelets, and detail the influence of the parameters used to simplify the monovariate functions. We provide the reconstruction quality that can be achieved and present the image reconstructions generated for each three considered simplification.

2. ALGORITHM

We briefly describe the algorithm proposed by Igelnik, and we invite the interested reader to refer to Ref. 3 and 4 for a detailed description of the algorithm. The first step is the definition of a disjoint tilage over the definition space $[0,1]^d$ of the multivariate function f. To entirely cover the space, several tilage layers are generated by translation of the first layer, as illustrated in Figure 1. For a given tilage layer n, d inner functions ψ_{ni} are randomly generated: one per dimension, independently from function f. The functions ψ_{ni} are sampled with M points, that are interpolated by cubic splines. The convex combination of these internal functions ψ_{ni} with real, linearly independent, and strictly positive values λ_i is the argument of external function g_n (one per dimension). Finally, the external function is constructed, using multivariate function values at the centers of hypercubes. To optimize network construction, each layer is weighted by coefficients a_n and summed to approximate the multivariate function f.

With this scheme, the original equation 1.1 becomes:

$$f(x_1,...,x_d) \simeq \sum_{n=1}^{N} a_n g_n \left(\sum_{i=1}^{d} \lambda_i \psi_{ni}(x_i) \right). \tag{2}$$

REMARK 1. *In the equation 1.1, one internal function ψ is defined for the whole network, and the argument x_i is translated for each layer n of a constant b_n. In this algorithm, one inner function ψ is defined per dimension (index i) and layer (index n).*

The tilage is then constituted with hypercubes H_n (the tiles) obtained by cartesian product of the intervals $I_n(j)$, defined as follows:

DEFINITION 2.1.

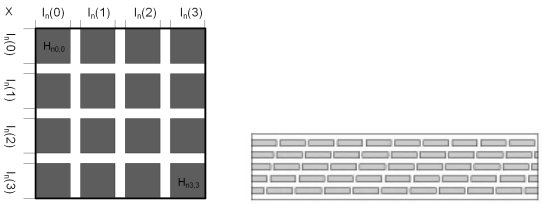
$$\forall n \in [\![1,N]\!], j \geqslant -1, I_n(j) = [(n-1)\delta + (N+1)j\delta, (n-1)\delta + (N+1)j\delta + N\delta],$$

where δ is the distance between two intervals I of length $N\delta$, such that the function f oscillation is smaller than $\frac{1}{N}$ on each hypercube H. j is an index that gives the position of an interval I_n. Values of j are defined such that the previously generated intervals $I_n(j)$ intersect the interval $[0,1]$, as illustrated in Figure 1.

2.1 Inner functions construction ψ_{ni}

Each function ψ_{ni} is defined as follows:

- generate a set of j distinct numbers y_{nij}, between Δ and $1-\Delta$, $0 < \Delta < 1$, such that the oscillations of the interpolating cubic spline of ψ values on the interval δ is lower than Δ.
- j is given by the intersection condition of intervals $I_n(j)$ with $[0,1]$ (see definition 2.1).



(a) (b)

Figure 1. (a) Cartesian product of intervals I to define a disjoint tilage of hypercubes C. (b) Superposition of translated disjoint tilages.

- The real numbers y_{nij} are sorted, *i.e.*: $y_{nij} < y_{nij+1}$. The image of the interval $I_n(j)$ by function ψ is y_{nij}.

This discontinuous inner function ψ is sampled by M points, that are interpolated by a cubic spline, as illustrated in Figure 3(a). We obtain two sets of points: points located on plateaus over intervals $I_n(j)$, and points M' located between two intervals $I_n(j)$ and $I_n(j+1)$, that are randomly placed. Points M' are optimized during the network construction, using a stochastic approach (see Ref. 3).

Once functions ψ_{ni} are constructed, the argument $\sum_{i=1}^{d} \lambda_i \psi_{ni}(x)$ of external functions can be evaluated. On hypercubes $H_{nij_1,...,j_d}$, it has constant values $p_{nj_1,...,j_d} = \sum_{i=1}^{d} \lambda_i y_{nij_i}$. Every random number y_{nij_i} generated verifies that the generated values $p_{nij_1,...,j_d}$ are all different, $\forall i \in [\![1,d]\!], \forall n \in [\![1,N]\!], \forall j \in \mathbb{N}, j \geqslant -1$. The real numbers λ_i, must be chosen linearly independent, strictly positive, and such that $\sum_{i=1}^{d} \lambda_i \leqslant 1$.

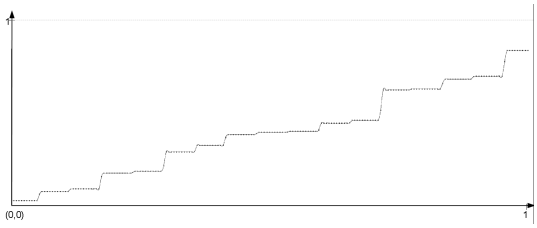
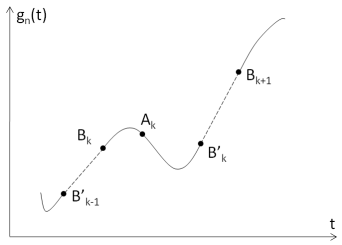


Figure 2. Example of function ψ sampled by 500 points that are interpolated by a cubic spline.

2.2 External function construction g_n

The function g_n is defined as follows:

- For every real number $t = p_{n,j_1,...,j_d}$, function $g_n(t)$ is equal to the N^{th} of values of the function f at the center of the hypercube $H_{nij_1,...,j_d}$, noted A_k.
- The definition interval of function g_n is extended to all $t \in [0,1]$. Two points B_k and B'_k are placed in A_k neighborhood, such that $t_{B_k} < t_{A_k} < t_{B'_k}$. The placement of points B_k and B'_k in the circles centered in A_k must preserve the order of points: $..., B'_{k-1}, B_k, A_k, B'_k, B_{k+1}, ...$, *i.e.* the radius of these circles must be smaller than half of the length between two consecutive points A_k. Points B'_k and B_{k+1} are connected with a line defined with a slope r. Points A_k and B'_k are connected with a nine degree spline s, such that: $s(t_{A_k}) = g_n(t_{A_k})$, $s(t_{B'_k}) = g_n(t_{B'_k})$, $s'(t_{B'_k}) = r$, and $s^{(2)}(t_{B'_k}) = s^{(3)}(t_{B'_k}) = s^{(4)}(t_{B'_k}) = 0$. Points B_k and A_k are connected with a similar nine degree spline. The connection condition at points A_k of both nine degree splines give the remaining conditions. This construction ensures the function continuity and the convergence of the approximating function to f (proved in Ref. 3).



(a)

Figure 3. (a) Plot of g_n. Points B, A and B' are connected by a nine degree spline. Points B' and B are connected by lines.

2.3 Network stochastic construction

The construction of monovariate functions requires some parameters to be optimized using a stochastic method (ensemble approach, see Ref. 4): the weights a_n associated to each layer, and the placement of the sampling points M' of inner functions ψ that are located between two consecutive intervals. To optimize the network convergence, three sets of points are constituted: a training set D_T, a generalization set D_G, and a validation set D_V. N layers are successively built. To add a new layer, K candidate layers are generated with the same plateaus y_{nij}, which gives K new candidate networks. The difference between two candidate layers is the set of sampling points M' located between two intervals $I_n(j)$ and $I_n(j+1)$, that are randomly chosen. We keep the layer from the network with the smallest mean squared error that is evaluated using the generalization set D_G. The weights a_n are obtained by minimizing the difference between the approximation given by the network and the image of function f for the points of the training set D_T. The algorithm is iterated until N layers are constructed. The validation error of the final network is determined using validation set D_V, *i.e.* by applying the approximated function to D_V.

To determine coefficients a_n, the difference between f and its approximation $\tilde{f}$ must be minimized:

$$\|Q_n a_n - t\|, \ noting \ t = \begin{bmatrix} f(x_{1,1},...,x_{d,1}) \\ ... \\ f(x_{1,P},...,x_{d,P}) \end{bmatrix}, \tag{3}$$

with Q_n a matrix of column vectors $q_k, k \in [\![0,n]\!]$ that corresponds to the approximation ($\tilde{f}$) of the k^{th} layer for points set $\big((x_{1,1},...,x_{d,1}),...,(x_{1,P},...,x_{d,P})\big)$ of D_T:

$$Q_n = \Big[\begin{bmatrix} \tilde{f}_0(x_{1,1},...x_{d,1}) \\ ... \\ \tilde{f}_0(x_{1,P},...x_{d,P}) \end{bmatrix}, ..., \begin{bmatrix} \tilde{f}_n(x_{1,1},...x_{d,1}) \\ ... \\ \tilde{f}_n(x_{1,P},...x_{d,P}) \end{bmatrix} \Big].$$

An evaluation of the solution $Q_n^{-1}t = a_n$ is proposed by Igelnik in Ref. 4. The coefficient a_l of the column vector $(a_0,...,a_n)^T$ is the weight associated to layer l, $l \in [\![0,n]\!]$. Figure 4 presents an overview of a network constituted of 5 tilage layers.

3. RESULTS

Using the algorithm presented in the previous section, we can decompose gray level images (seen as bivariate functions) into monovariate functions. Each pixel corresponds to a tile of the bidimensional space $[0,1]^d$, where the bivariate function has a constant value. In Ref. 5, the authors have shown that the combination of KST decomposition with a multiresolution approach improves the reconstruction quality. A wavelet decomposition leads to 4 sub-images, one is a low-frequencies image, and three contains high frequencies. Our goal is to decompose the images of details and, taking advantage of the limited contrast, to replace values in external functions to reduce the number of pixels from the original image required to the external functions construction. We propose three simplification methods.

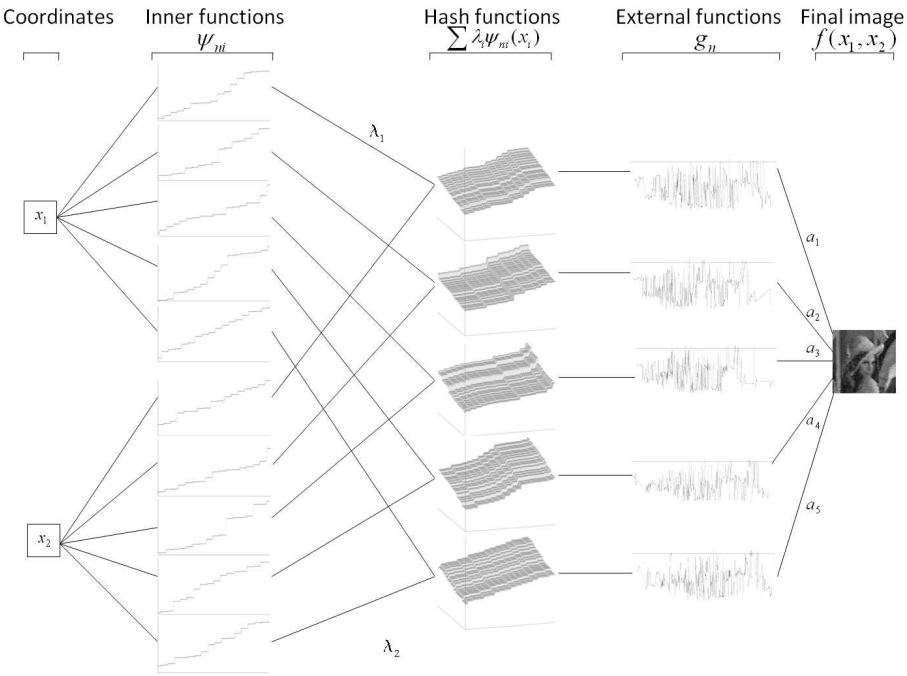
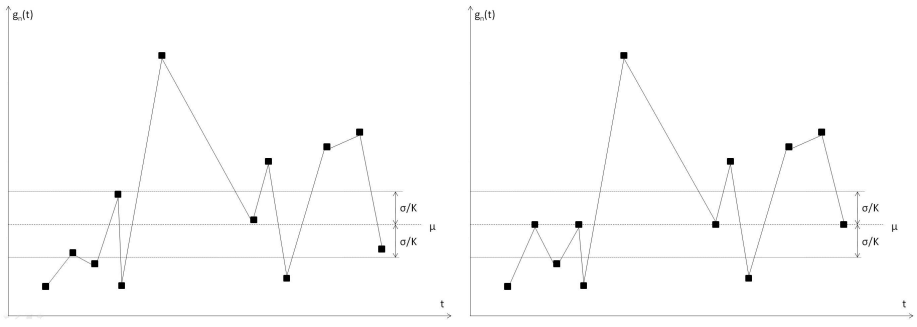


Figure 4. Overview of a 5 tilage layer network.

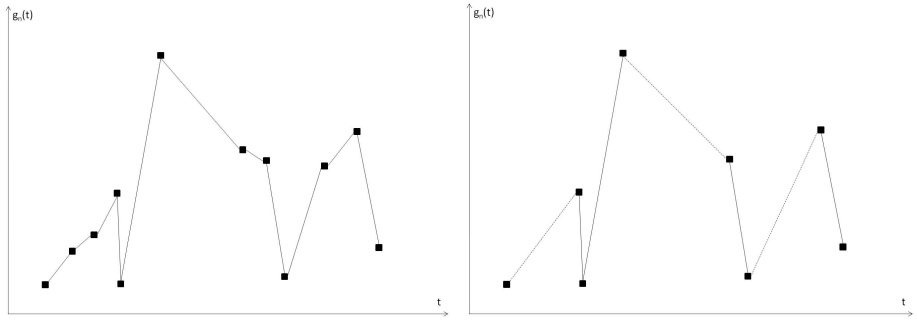
The first one is related to tilage definition. By changing the parameter δ, the size of the tilage can be adjusted, *i.e.* the number of tiles per layer. The tile size directly determines the number of pixels of the image of details that are utilized for the network construction: only pixel values located on the center of tiles are utilized to construct external functions g_n. Decreasing the number of pixels from the original image in external functions (*i.e.* increasing tile size) leads to a partial use of the original image pixels.
The second approach uses a fixed tilage size and it is applied during the construction of the external function. The first step of this construction is the generation of points corresponding to the pixels located at the center of the hypercubes (values with abscissa $p_{n,j_1,\ldots,j_d}$, see section 2.2). The simplification is then applied after this first step to decrease the number of pixels retained for the network construction. The mean value of the external function is computed, and the points located at a distance from the mean value smaller than the standard deviation are replaced by the mean value of the external function. In other words, the mean value μ_n and standard deviation σ_n of the external function are computed, and every value such that $K \in \mathbb{N}^*, |g_n - \mu_n| < \frac{\sigma_n}{K}$ is replaced by μ_n. The constant K can be adjusted: the smaller, the more points are simplified. With this method, the pixels located at the center of the hypercubes and used to compute these points are no more required after simplification. Figure 5 illustrates such a simplification: (a) is the original external function, and (b) is the external function after simplification.
The third approach takes advantage of the external function continuity. As the mean value simplification, it is applied during external function construction, after the generation of values corresponding to the centers of the hypercubes. Some values in external functions can be linearly interpolated without losing any information. The pixels used to compute these points in the external functions are not required anymore and can be ignored. Precisely, each point of abscissa t_0 such that $\epsilon \in \mathbb{R}, |\frac{g(t_0)-g(t_{-1})}{t_0-t_{-1}} - \frac{g(t_1)-g(t_{-1})}{t_1-t_{-1}}| \leqslant \epsilon$ is removed. The constant ϵ can be adjusted: the greater, the more points are simplified. Figure 6 presents an example of this simplification: points in (a) verifying the alignment criteria have been removed in (b).

We evaluate the efficiency of these three compression approaches. All measures are performed using a 384×384 pixels Lena and a 10-layer networks ($N = 10$). We apply each simplification method and study the PSNR of the reconstructed image as a function of the number of pixels utilized to define external functions. Figure 7 presents the results. By changing the tilage density (δ), the reconstruction quality rapidly decreases and tends to the reconstruction obtained using only the low-frequencies image. This direct approach cannot be used for



(a) (b)

Figure 5. Simplification of external function using the mean value. (a) Original function. (b) function after simplification.



(a) (b)

Figure 6. Simplification of external function using linear interpolation. (a) Original function. (b) function after simplification.

compression, which leads us to consider the two approaches for external function simplifications. To study these two simplifications, the tilage size has been fixed to use 100% of the pixels of the image. The best compression performance is obtained with the simplification using mean values. Using very low compression rates (0-5%), the reconstruction quality provided by the simplification by linearity is almost as good as the simplification provided by mean values: as a research perspective, a combination of these two simplifications could be computed. Low compression of linear simplification could be associated to the mean value simplification to improve the compression efficiency.

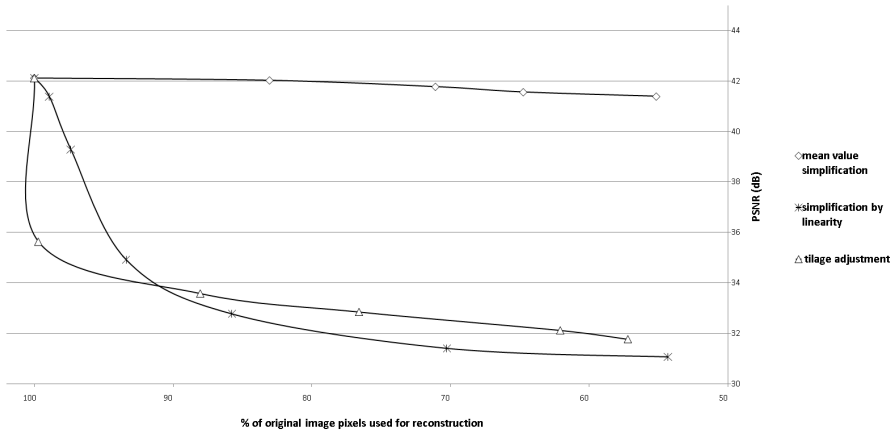


Figure 7. PSNR of image reconstruction with the 3 possible approaches as a function of the number of pixels utilized to define external functions.

Figure 8 presents the three reconstructions obtained using the three compression approaches with a rate of

55%. We can note that the difference of reconstruction accuracy between the approaches is only related to the quantity of details that are preserved or not, since no artifacts are generated.

(a) (b) (c)

Figure 8. Reconstructions of Lena at around 55% compression rate, using: (a) tilage definition, (b) mean value simplification and (c) simplification by linearity.

4. CONCLUSION AND PERSPECTIVES

We have dealt with multivariate function decomposition using KST. We have presented our implementation of Igelnik's algorithm, that provides control over the size of tiles, which determines the quantity of pixels from the decomposed image that are utilized and contained in the network. Using this size reduction, we have proposed a compression approach, that has been proved adapted to the decomposition of subimages of details obtained from a wavelet decomposition. Due to the simple representation of the high frequencies, the monovariate functions can be simplified: the three images of details can be replaced by a decomposition into simplified monovariate functions, preserving the reconstruction quality. We have presented two simplification methods of these monovariate functions, and studied their influence on the final image reconstruction.

Our principal contribution is the presentation and the characterization of a new compression approach, combining KST decomposition, wavelets decomposition, and three different monovariate function simplifications: the decomposition of an image into continuous monovariate functions associated to a superposition of tilage layers can be used to compress an image. Moreover, the reconstruction and compression rate improved by applying this decomposition to wavelet image decomposition. We have proposed three different approaches to obtain simpler monovariate functions and compress the image, and detailed the results obtained for each approach.

Several research perspectives can be pointed out: further developments of this approach are being conducted on the combination of the simplifications, and on computation of a complete compression approach by inserting this approach into JPEG 2000 to obtain a complete compression scheme. The second perspective is the addition of encryption and authentication to this compression scheme: considering the direct decomposition of an image into monovariate functions, one can remark that definitions of external and internal monovariate functions are independent. Moreover, internal functions are required to re-arrange external functions and reconstruct the image. Can internal functions be used as a signature or as an encryption key?

REFERENCES

[1] Sprecher, D. A., "A numerical implementation of Kolmogorov's superpositions," *Neural Networks* **9**(5), 765–772 (1996).

[2] Sprecher, D. A., "A numerical implementation of Kolmogorov's superpositions ii," *Neural Networks* **10**(3), 447–457 (1997).

[3] Igelnik, B. and Parikh, N., "Kolmogorov's spline network," *IEEE transactions on neural networks* **14**(4), 725–733 (2003).

[4] Igelnik, B., Pao, Y.-H., LeClair, S. R., and Shen, C. Y., "The ensemble approach to neural-network learning and generalization," *IEEE Transactions on Neural Networks* **10**, 19–30 (1999).

[5] Leni, P.-E., Fougerolle, Y. D., and Truchetet, F., "Kolmogorov superposition theorem and its application to wavelet image decompositions," *Electronic Imaging - Wavelet Applications in Industrial Processing VI, Proceedings of the SPIE* **7248**, 724804–724804–12 (2009).

[6] Igelnik, B., "Kolmogorov's Spline Complex Network and Adaptive Dynamic Modeling of Data," *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters* , 56 (2009).

[7] Köppen, M., "On the training of a Kolmogorov Network," *Lecture Notes in Computer Science, Springer Berlin* **2415**, 140 (2002).

[8] Köppen, M. and Yoshida, K., "Universal representation of image functions by the sprecher construction," *Soft Computing as Transdisciplinary Science and Technology, Springer Berlin* **29**, 202–210 (2005).

[9] Brattka, V., "Du 13-ième problème de Hilbert à la théorie des réseaux de neurones : aspects constructifs du théorème de superposition de Kolmogorov," *L'héritage de Kolmogorov en mathématiques. Éditions Belin, Paris.* , 241–268 (2004).

[10] Braun, J. and Griebel, M., "On a constructive proof of Kolmogorov's superposition theorem," *Constructive approximation* (2007).

[11] Hecht-Nielsen, R., "Kolmogorov's mapping neural network existence theorem," *Proceedings of the IEEE International Conference on Neural Networks III, New York* , 11–13 (1987).

[12] Igelnik, B., Tabib-Azar, M., and LeClair, S. R., "A net with complex weights," *IEEE Transactions on Neural Networks* **12**, 236–249 (2001).

[13] Lagunas, M. A., Pérez-Neira, A., Nájar, M., and Pagés, A., "The Kolmogorov Signal Processor," *Lecture Notes in Computer Science, Springer Berlin* **686**, 494–512 (1993).

[14] Moon, B., "An explicit solution for the cubic spline interpolation for functions of a single variable," *Applied Mathematics and Computation* **117**, 251–255 (2001).