



Kolmogorov Superposition Theorem and Wavelet Decomposition for Image Compression

Pierre-Emmanuel Leni, Yohan Fougerolle, Frederic Truchetet

► To cite this version:

Pierre-Emmanuel Leni, Yohan Fougerolle, Frederic Truchetet. Kolmogorov Superposition Theorem and Wavelet Decomposition for Image Compression. Advanced Concepts for Intelligent Vision Systems, Sep 2009, Bordeaux, France. pp.43-53. hal-00634071

HAL Id: hal-00634071

<https://hal.science/hal-00634071>

Submitted on 20 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Kolmogorov Superposition Theorem and Wavelet Decomposition 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. Kolmogorov Superposition Theorem stands that any multivariate function can be decomposed into two types of monovariate functions that are called inner and external functions: each inner function is associated to one dimension and linearly combined to construct a hash-function that associates every point of a multidimensional space to a value of the real interval $[0, 1]$. These intermediate values are then associated by external functions to the corresponding value of the multidimensional function. Thanks to the decomposition into monovariate functions, our goal is to apply this decomposition to images and obtain image compression.

We propose a new algorithm to decompose images into continuous monovariate functions, and propose a compression approach: thanks to the decomposition scheme, the quantity of information taken into account to define the monovariate functions can be adapted: 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. To improve the reconstruction quality, we combine KST and multiresolution approach, 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: we combine KST and multiresolution approach. Taking advantage of the KST decomposition scheme, the low frequencies will be represented with the highest accuracy, and the high frequencies representation will be replaced by a decomposition into simplified monovariate functions, preserving the reconstruction quality. We detail our approach and our results on different images and present the reconstruction quality as a function of the quantity of pixels contained in monovariate functions.

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 [11], [12], can be written as:

Theorem 1 (Kolmogorov superposition theorem). *Every continuous function defined on the identity hypercube, $f : [0, 1]^d \rightarrow \mathbb{R}$, can be written as sums and compositions of continuous univariate functions as:*

$$f(x_1, \dots, x_d) = \sum_{n=0}^{2d} g_n \left(\sum_{i=1}^d \lambda_i \psi(x_i + b_n) \right), \quad (1)$$

with ψ continuous function, λ_i and b constants. ψ is called inner function and g external function.

Coordinates $x_i, i \in \llbracket 1, d \rrbracket$ 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 .

Igel'nik has presented in [5] an approximating construction that provides flexibility and modification perspectives over the univariate function construction. Using Igel'nik'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 univariate functions containing only a fraction of the original image pixels. To improve the reconstruction quality, we apply this decomposition on images of details obtained by a wavelet decomposition: external functions obtained from the decomposition of images of details can be simplified, providing better reconstruction quality than larger tile sizes.

The structure of the paper is as follows: we present the decomposition algorithm in section 2. In section 3, we present the results of gray level image decompositions, and combine the KST decomposition with wavelets to improve the reconstruction. In the last section, we present our conclusions and several promising research perspectives.

Our contributions include improvements and modifications of Igel'nik's algorithm for image decomposition, the characterization of the obtained continuous decomposition, and the determination of the reconstruction quality as a function of the quantity of information contained in univariate functions.

2 Algorithm

We briefly describe the algorithm proposed by Igel'nik, and we invite the interested reader to refer to [5] and [4] for a detailed description of the algorithm. The first step is the definition of a disjoint tiling over the definition space $[0, 1]^d$ of the multivariate function f . To entirely cover the space, several tiling layers are generated by translation of the first layer, as illustrated in Figure 1. For a given tiling 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 becomes:

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

Remark 1. In the equation 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 C_n obtained by cartesian product of the intervals $I_n(j)$, defined as follows:

Definition 1.

$$\forall n \in \llbracket 1, N \rrbracket, j \geq -1, I_n(j) = [(n-1)\delta + (N+1)j\delta, (n-1)\delta + (N+1)j\delta + N\delta],$$

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 C . Values of j are defined such that the previously generated intervals $I_n(j)$ intersect the interval $[0, 1]$, as illustrated in Figure 1.

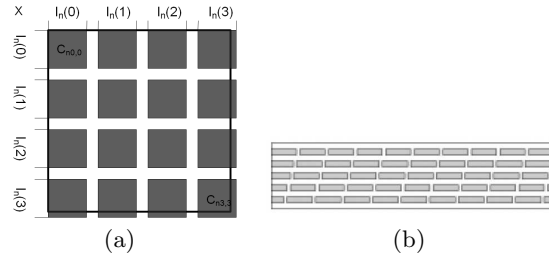


Fig. 1. (a) Cartesian product of intervals I to define a disjoint tilage of hypercubes C . (b) Superposition of translated disjoint tilages.

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 definition 1. 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 [5]).

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 C_{nij_1, \dots, j_d} , it has constant values $p_{nj_1, \dots, j_d} = \sum_{i=1}^d \lambda_i y_{nij_i}$. Every random number y_{nij_i} generated verifies that the generated values p_{nij_i} are all different, $\forall i \in \llbracket 1, d \rrbracket, \forall n \in \llbracket 1, N \rrbracket, \forall j \in \mathbb{N}, j \geq -1$.

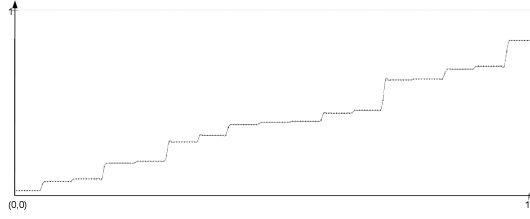


Fig. 2. Example of function ψ sampled by 500 points that are interpolated by a cubic spline.

2.2 External function construction g_n

The function g_n is defined as follows:

- For every real number $t = p_{n,j_1, \dots, j_d}$, function $g_n(t)$ is equal to the N^{th} of values of the function f at the center of the hypercube C_{nij_1, \dots, 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: $\dots, B'_{k-1}, B_k, A_k, B'_k, B_{k+1}, \dots$, *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 [5]). Figure 3(b) illustrates this construction.

The external function has a "noisy" shape, which is related to the global sweeping scheme of the image: Sprecher and *al.* have demonstrated in Ref.[13]

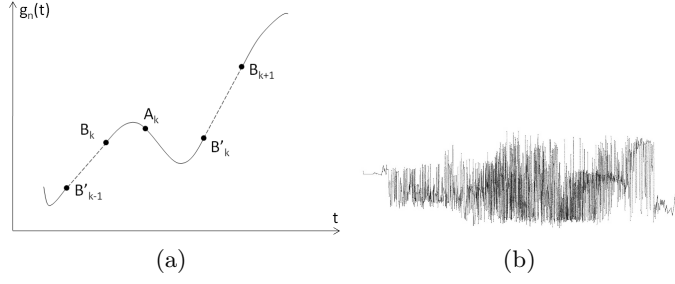


Fig. 3. (a) Plot of g_n . Points B , A and B' are connected by a nine degree spline. Points B' and B are connected by lines. (b) Example of function g_n for a complete layer of Lena decomposition.

that using internal functions, space-filling curves can be defined. The linear combination of inner functions associates a unique real value to every couple of the multidimensional space $[0, 1]^d$. Sorting these real values defines a unique path through the tiles of a layer: the space filling curve. Figure 4 illustrates an example of such a curve: the pixels are swept without any neighborhood property conservation.

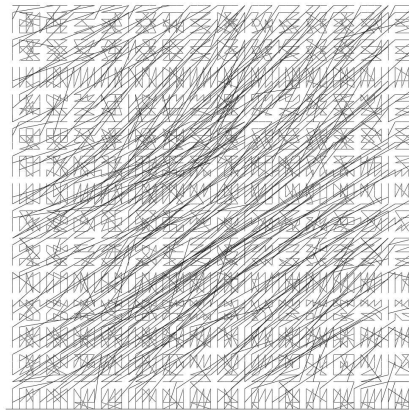


Fig. 4. Igel'nik's space filling curve.

2.3 Network stochastic construction

The construction of monovariate functions requires some parameters to be optimized using a stochastic method (ensemble approach, see [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:

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

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

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

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 $(a_0, \dots, a_n)^T$ is the weight associated to layer $l, l \in \llbracket 0, n \rrbracket$. Figure 5 presents an overview of a network constituted of 5 tilage layers.

3 Results

The algorithm presented in the previous section can be used to decompose gray level images (seen as bivariate functions). Each pixel corresponds to a tile of the bidimensional space $[0, 1]^d$, where the bivariate function has a constant value. By changing the parameters δ and N , 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 original image that are utilized for the network construction: 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. To characterize the compression properties of the network, we represent the number of pixels of the original image that are contained in the network as a function of PSNR reconstruction, by training the network with a 100x100 pixels image

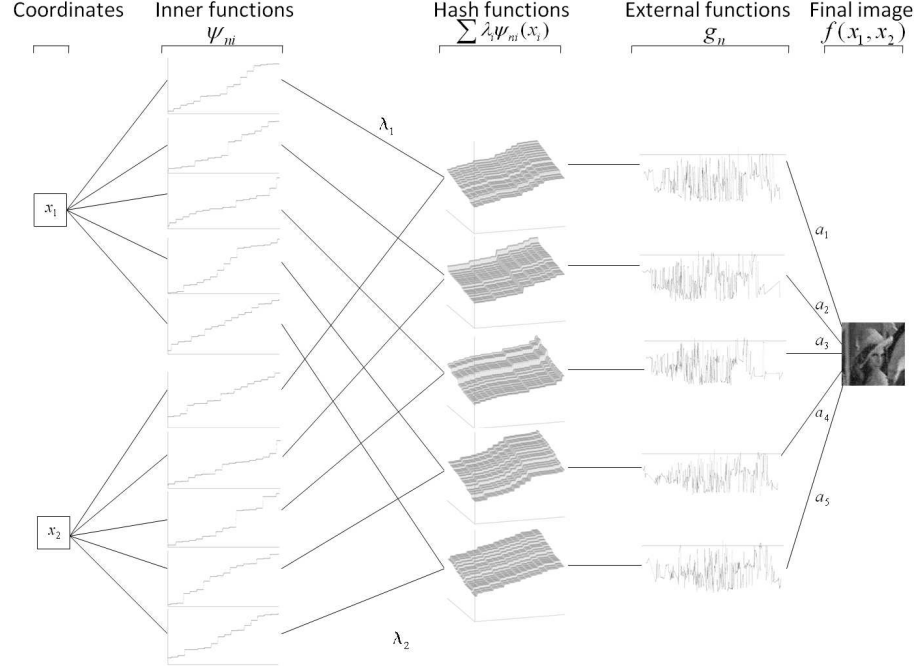


Fig. 5. Overview of a 5 tilage layer network.

to reconstruct a 100x100 pixels image. Figure 6 presents several reconstructions obtained using between 100% and 15% of the pixels from original image, and figure 8 (dot line) details the obtained PSNR. Figure 7 presents the reconstruction using KST and reconstructions obtained using bicubic and nearest neighbor interpolation of an image containing only a fraction of original image pixels.



Fig. 6. Lena reconstruction, using 100%(a), 70%(b), 40%(c), 15%(d) of original image pixels.

We combine our decomposition scheme with a multiresolution approach to improve the reconstruction quality: a wavelet decomposition leads to 4 sub-

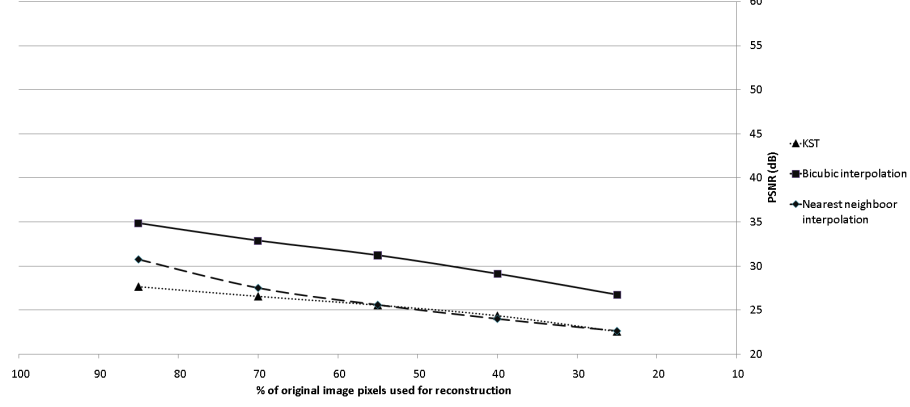


Fig. 7. PSNR of image reconstruction as a function of the number of pixels utilized to define external functions.

images, one is a low-frequencies image, and three contains high frequencies. Our goal is to decompose the images of details using small tiles, 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 (as if larger tiles were used). An interesting property of this approach is that the decomposition of the images of details (high frequencies) leads to simple external functions, with limited oscillations. Computing the mean value for every external function and replacing the values located at a distance from the mean value smaller than the standard deviation allows to decrease the number of pixels retained for the network construction. Precisely, the smallest tile size (a tile = a pixel) is utilized to decompose an image of details, and is reduced up to only 15% of pixels after external function simplification. We compare this simplification approach with well known image interpolation techniques: we reconstruct 100x100 images of details obtained with wavelets decomposition using bicubic and nearest neighbor interpolation with only a fraction of the original image pixels. Figure 8 details the obtained PSNR for Lena: the reconstruction PSNR is higher and above 30dB for up to a 60% compression rate, and no visible artefacts can be seen (see figure 10). Figure 9 presents the compression and associated reconstruction PSNR for five images: Lena, Goldhill, Peppers, Barbara, and Mandrill. The reconstruction PSNR is higher and above 40 dB for up to 65% compression rate. Figure 10 presents the results obtained on these five images with two external functions simplification (high and low compression rates). We observe that the reconstruction is not visibly altered. The irregular repartition of measures is due to the external function simplifications: measures are realized with regular simplification criteria over external functions, but the simplification of external functions obtained is image dependent, so the compression ratio.

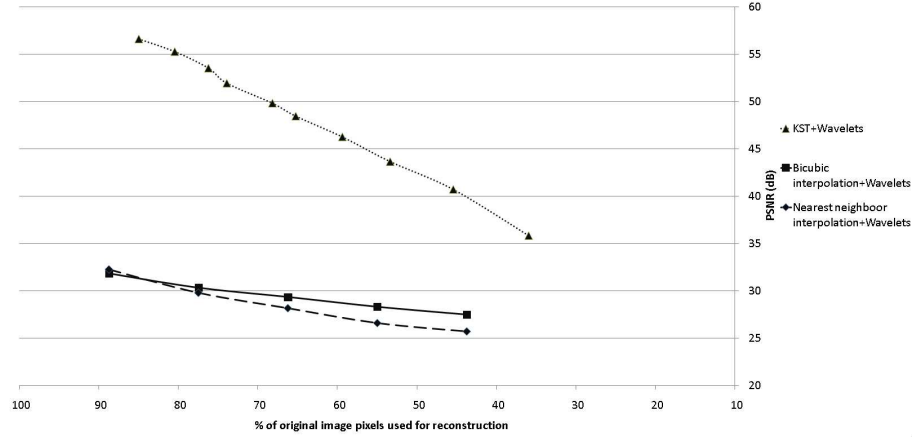


Fig. 8. PSNR of image reconstruction using wavelets as a function of the number of pixels utilized to define external functions.

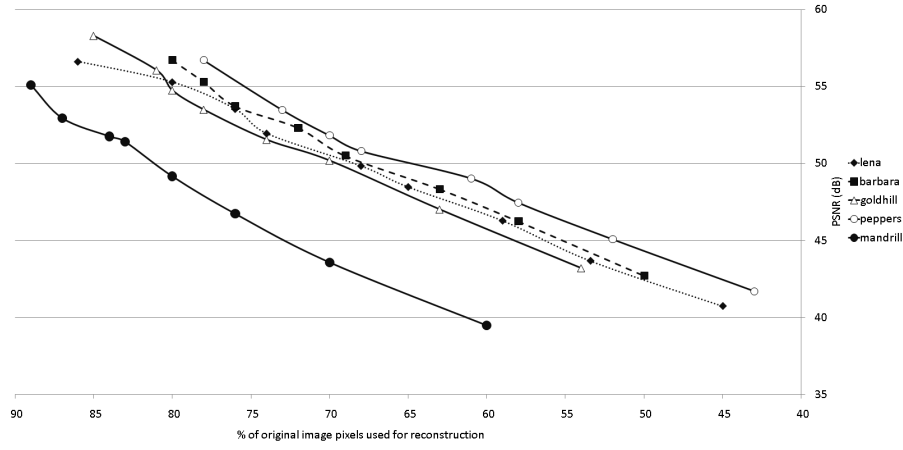


Fig. 9. PSNR of five classical image reconstruction.

4 Conclusion and perspectives

We have dealt with multivariate function decomposition using KST. We have presented our implementation of Igel'nik'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 de-

composition into simplified monovariate functions, preserving the reconstruction quality. We have presented the results of our approach, applied to various gray level images and different simplification parameters of external functions.

Our principal contribution is the presentation of a compression method, combining KST decomposition and wavelets decomposition simplification: the decomposition of an image into continuous monovariate functions, that relies on a superposition of tilage layers, that can be used to compress an image, and the reconstruction and compression rate improved by applying this decomposition to wavelet image decomposition.

From these results, several perspectives can be pointed out: further developments of this approach are required to obtain a complete compression method, *i.e.* the size of the compressed image has to be evaluated, which implies the development of an adapted quantization. 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? And finally, a multiresolution approach can be considered: layers of a network can have different tilage densities, so the image could be progressively reconstructed, with an increasing resolution, by progressively superposing the layers.

References

1. Vasco Brattka. 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.*, pages 241–268, 2004.
2. Jürgen Braun and Michael Griebel. On a constructive proof of Kolmogorov's superposition theorem. *Constructive approximation*, 2007.
3. Robert Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. *Proceedings of the IEEE International Conference on Neural Networks III, New York*, pages 11–13, 1987.
4. Boris Igelnik, Yoh-Han Pao, Steven R. LeClair, and Chang Yun Shen. The ensemble approach to neural-network learning and generalization. *IEEE Transactions on Neural Networks*, 10:19–30, 1999.
5. Boris Igelnik and Neel Parikh. Kolmogorov's spline network. *IEEE transactions on neural networks*, 14(4):725–733, 2003.
6. Boris Igelnik, Massood Tabib-Azar, and Steven R. LeClair. A net with complex weights. *IEEE Transactions on Neural Networks*, 12:236–249, 2001.
7. Mario Köppen. On the training of a Kolmogorov Network. *Lecture Notes in Computer Science, Springer Berlin*, 2415:140, 2002.
8. Miguel A. Lagunas, Ana Pérez-Neira, Montse Nájjar, and Alba Pagés. The Kolmogorov Signal Processor. *Lecture Notes in Computer Science, Springer Berlin*, 686:494–512, 1993.
9. B.S. Moon. An explicit solution for the cubic spline interpolation for functions of a single variable. *Applied Mathematics and Computation*, 117:251–255, 2001.

10. David A. Sprecher. An improvement in the superposition theorem of Kolmogorov. *Journal of Mathematical Analysis and Applications*, 38:208–213, 1972.
11. David A. Sprecher. A numerical implementation of Kolmogorov’s superpositions. *Neural Networks*, 9(5):765–772, 1996.
12. David A. Sprecher. A numerical implementation of Kolmogorov’s superpositions ii. *Neural Networks*, 10(3):447–457, 1997.
13. David A. Sprecher and Sorin Draghici. Space-filling curves and Kolmogorov superposition-based neural networks. *Neural Networks*, 15(1):57–67, 2002.



Fig. 10. Reconstruction of (a)(b) lena, (c)(d) barbara, (e)(f) goldhill, (g)(h) peppers and (i)(j) mandrill, using about 50% and 80% (respectively) of the number of original image pixels.