Nonlocal Multiscale Hierarchical Decomposition on Graphs

Moncef Hidane, Olivier Lezoray, Vinh Thong Ta, Abderrahim Elmoataz

Nonlocal Multiscale Hierarchical Decomposition on Graphs

Moncef Hidane, Olivier Lézoray, Vinh-Thong Ta, and Abderrahim Elmoataz

Université de Caen Basse-Normandie, ENSICAEN, CNRS, GREYC Image Team,
6 Boulevard Maréchal Juin, F-14050 Caen Cedex France.
{moncef.hidane,olivier.lezoray,vinhthong.ta,abderrahim.elmoataz-billah}@unicaen.fr

Abstract. The decomposition of images into their meaningful components is one of the major tasks in computer vision. Tadmor, Nezzar and Vese [1] have proposed a general approach for multiscale hierarchical decomposition of images. On the basis of this work, we propose a multiscale hierarchical decomposition of functions on graphs. The decomposition is based on a discrete variational framework that makes it possible to process arbitrary discrete data sets with the natural introduction of nonlocal interactions. This leads to an approach that can be used for the decomposition of images, meshes, or arbitrary data sets by taking advantage of the graph structure. To have a fully automatic decomposition, the issue of parameter selection is fully addressed. We illustrate our approach with numerous decomposition results on images, meshes, and point clouds and show the benefits.

1 Introduction

It is well-accepted by now that vision is inherently a multiscale phenomenon. A visual task comes to representing and interpreting visual scenes with its singularities. As a consequence, a mathematical multiscale representation is essential to analyze images and decomposing an image into its meaningful components is one of the major tasks in computer vision and image processing. Images usually contain two types of main components: structure and texture. Structure is mainly the sharp edges in the image, which separate different objects. Texture is in general a repetitive pattern or the presence of oscillations (that can be noise). A typical image decomposition problem that is often considered is image denoising [2] where an image f is separated into a clean part u and a noisy part v. A popular scheme that proposed such a decomposition was the total variation decomposition of Rudin, Osher and Fatemi (ROF) [3] which was introduced as an effective denoising tool:

$$\inf_u \left\{ \int_\Omega |\nabla u| + \lambda \|v\|_{L^2}^2, f = u + v \right\} \tag{1}$$

where λ is a scale parameter. With the ROF model, as pointed out by Meyer [2], image denoising leads to image decomposition where the minimization of the

previous functional leads to a decomposition of f into a part u that extracts the edges of f, and a part v that captures the texture. Moreover, denoising at different scales λ generates a multiscale representation. In [1], Tadmor, Nezzar and Vese proposed a new multiscale image decomposition which offers a hierarchical and adaptive representation for different features in images. An image is hierarchically decomposed into the sum of simpler atoms u_k, where u_k extracts more refined information from the previous scale u_{k-1}. To this end, the atoms u_k are obtained as dyadically scaled minimizers of the ROF functional at increasing λ_k scales. Recently, the integration of nonlocal interactions in images [4] has shown to be very effective for capturing repetitive patterns in images and has led to numerous developments for image denoising (see [5] an reference therein). As it has been shown by Elmoataz *et al.* in [6], the formalism of difference equations on graphs provides a unifying view of local and nonlocal processing of functions on graphs (from images to high dimensional data). However, the integration of nonlocal interactions [7] has been few investigated for image decomposition.

Contributions. In this work, we propose to generalize the multiscale hierarchical decomposition of Tadmor, Nezzar and Vese (TNV) [1] proposed for images to a multiscale hierarchical decomposition of functions on arbitrary graphs. This naturally enables, first, to integrate nonlocal information in the decomposition process and second, to generalize it to the decomposition of functions on graphs. With the proposed approach, it is then possible to obtain a decomposition integrating nonlocal information for images, point clouds, or meshes (see [8] for an alternative aproach) once a graph representation is associated to the data under consideration.

Paper organization. In Section 2, we recall the multiscale hierarchical decomposition of images introduced in [1]. The formulation in this latter section is continuous. In Section 3, we introduce the discrete variational of [6] and give the numerical algorithms used for the solution. We then move on to the extension of the multiscale hierarchical decomposition to general data lying on graphs using the former framework. The approaches taken to construct weighted graphs are given. We then address the issue of parameter selection and give a general selection procedure. In the last Section, we illustrate the capabilities of the decomposition by applying it to different kinds of data defined on graphs. Grayscale, color, synthetic and natural images are considered as well as 3-dimensional meshes and point clouds.

2 Multiscale Hierarchical Decomposition of Images

The Tadmore Nezzar Vese multiscale hierarchical decomposition [1] builds upon the total variation minimization introduced in [3]. Given a noisy image f defined on continuous domain Ω, the authors in [3] consider the minimization (1) in order to recover the original denoised image. The starting point in [1] is an alternative point of view about the standard ROF denoising methodology. Inspired by the work of Meyer [2], Eq. (1) is interpreted as a decomposition: the

original noise-free image f is decomposed into a cartoon part u and an oscillatory or textured part v. The cartoon part should capture only the geometric features of the original image (i.e. sharp edges and boundaries) while the texture one should capture the repeated and meaningful small patterns. The trade-off between geometry and texture is dictated by the parameter λ which now plays the role of a scale separation parameter. The small meaningful patterns known as texture are in fact scale dependent: what might be seen as texture at a fixed scale could in fact contain important details when considered under a refined scale. To overcome this possible limitation, the authors in [1] propose to iterate the process in (1). Starting with an initial image f and an initial scale λ_0, a first decomposition is performed on f. Let u_0 be the regularized image and v_0 the residual so that $f = u_0 + v_0$. Then, another decomposition is performed, this time on the residual v_0 with a refined scale $\lambda_1 > \lambda_0$ leading to the following decomposition: $v_0 = u_1 + v_1$. We now have $f = u_0 + u_1 + v_1$. For a fixed index k and a sequence of refined scales $\lambda_0 > \lambda_1 > \ldots > \lambda_k$ we obtain the following decomposition:

$$f = u_0 + \ldots + u_k + v_k \ . \tag{2}$$

Results about the convergence as $k \longrightarrow \infty$ as well as upper and lower bounds for the choice of λ are provided in [1] for the geometric sequence $\lambda_i = 2^i$. Unfortunately, it is difficult to explicitly compute these bounds. Tadmor et al [1] argue that the choice of the parameter λ_0 is not important. If λ_0 is too small, the first decomposition levels, will be very smooth and details will not be reconstructed while λ_i is not large enough. While this is true for images with 4-adjacency grid graphs, it may be problematic for high dimensional data living on graphs mainly for two aspects: the dimension of the data can be high as well the nature of the processing (local or nonlocal). Therefore, a bad choice of λ_0 can produce higher computational cost for the decomposition since more levels are required. In [9], the hierarchical decomposition has been successfully applied to deblurring and denoising images as well as for segmentation.

3 Multiscale Hierarchical Decomposition on Graphs

In this Section, we recall the discrete variational framework introduced in [6] for the processing of general data defined on arbitrary graphs. Based on the ideas presented in [1], we use this framework to produce a multiscale decomposition of general data defined on graphs.

3.1 Digital Variational Framework

Definitions and Notations Let $G = (V, E, w)$ be a general weighted graph consisting of a set of vertices $V = \{\alpha_1, \ldots, \alpha_N\}$ and set of weighted edges $E \subset V \times V$ with a similarity measure $w : V \times V \rightarrow [0, 1]$. For an edge (α, β) connecting the vertices α and β, $w(\alpha, \beta)$ represents a similarity measure between the vertices often based on an *a priori* distance measure. w is supposed symmetric and satisfies $w(\alpha, \beta) = 0$ if $(\alpha, \beta) \notin E$. The graph G is assumed to be undirected,

with no self-loops and no multiple-edges.
Let $f : V \to \mathbb{R}$ be a real-valued function. To measure the regularity of f, the authors in [6] use the following difference operator df:

$$(df)(\alpha,\beta) \stackrel{def}{=} \partial_\beta f(\alpha) \stackrel{def}{=} \sqrt{w(\alpha,\beta)}(f(\beta)-f(\alpha)),\ \forall(\alpha,\beta)\in E\ . \tag{3}$$

The discrete gradient operator is then defined at vertex α as follows:

$$\begin{aligned}\nabla_w f(\alpha) &\stackrel{def}{=} [\partial_\beta f(\alpha)\ :\ \beta\sim\alpha]^T \\ &= [\partial_{\beta_1} f(\alpha),\ldots,\partial_{\beta_k} f(\alpha)]^T,\ \forall(\alpha,\beta_i)\in E\ .\end{aligned} \tag{4}$$

Its Euclidean norm represents a local measure of the variation of f at a given vertex:

$$\begin{aligned}|\nabla_w f(\alpha)| &= \sqrt{\sum_{\beta\sim\alpha}(\partial_\beta f(\alpha))^2} \\ &= \sqrt{\sum_{\beta\sim\alpha} w(\alpha,\beta)(f(\beta)-f(\alpha))^2}\ .\end{aligned} \tag{5}$$

Another important operator is the nonlocal curvature graph operator which is defined as follows [6]:

$$\kappa_w^f(\alpha) = \tfrac{1}{2}\sum_{\beta\sim\alpha}\gamma_w^f(\alpha,\beta)(f(\alpha)-f(\beta))\ , \tag{6}$$

$$\gamma_w^f(\alpha,\beta) = w(\alpha,\beta)(|\nabla_w f(\beta)|^{-1}+|\nabla_w f(\alpha)|^{-1})\ . \tag{7}$$

Discrete Energy Let f^0 be a real function defined on a weighted graph $G=(V,E,w)$. Usually f^0 is a corrupted version of a clean function g. The problem of recovering g from f^0 is a typical inverse problem. A standard method to solve this inverse problem is to consider it as a variational one. The variational formulation consists in the minimization of an energy functional involving a regularization term and a fidelity term. The energy considered here is :

$$E_w(f,f^0,\lambda) = \sum_{\alpha\in V}|\nabla_w f(\alpha)| + \tfrac{\lambda}{2}\|f-f^0\|^2\ , \tag{8}$$

where

$$\|f-f^0\|^2 = \sum_{\alpha\in V}|f(\alpha)-f^0(\alpha)|^2\ . \tag{9}$$

The first term of the right hand side in (8) is the regularization one, while the second one forces the solution to be closed to the initial data. The factor λ dictates the trade-off between these two terms. Then the minimization consists in finding:

$$\operatorname{arginf}\{E_w(f,f^0,\lambda),\ f:V\to\mathbb{R}\}\ . \tag{10}$$

For $w = 1$ the energy is the one proposed in [10], in the case of images modeled by 4 or 8-adjacency grid graphs. For $w \neq 1$ the formulation becomes an adaptive version of the digital TV filter [10]. If f^0 has its values in $\mathbb{R}^n$ then we consider an energy regularization for each component. In this case, the use of a global similarity measure plays the role of a correlation between the different channels and thus avoids the drawbacks of marginal processing [11].

3.2 Numerical Resolution

Restoration/Decomposition Equations In this Section, we derive the restoration/decomposition equations that will serve to decompose general data defined on graphs. The discrete formulation of the energies leads to a set of algebraic equations which are the discrete equivalent of the Euler-Lagrange equation. Since the energy (8) is strictly convex, the optimal solution is given by taking the derivative:

$$\frac{\partial E_w(f, f^0, \lambda)}{\partial f(\alpha)} = 0, \quad \forall \alpha \in V. \tag{11}$$

In practice, in order to avoid zero division, we replace the Euclidean norm of the discrete gradient operator by a regularized version:

$$|\nabla_w f(\alpha)|_\epsilon = \sqrt{|\nabla_w f(\alpha)|^2 + \epsilon^2}\ ,$$

where ϵ is a small fixed number, typically $\epsilon = 10^{-4}$. This amounts to consider the following energy function:

$$E_w(f, f^0, \lambda) = \sum_{\alpha \in V} |\nabla_w f(\alpha)|_\epsilon + \tfrac{\lambda}{2}\|f - f^0\|^2\ . \tag{12}$$

This regularized energy is still strictly convex, and thus the solution is again given by taking the derivatives. In the sequel, we will drop the ϵ subscript, and $|\nabla_w f|$ will mean $|\nabla_w f|_\epsilon$.
The restoration/decomposition equations are then written (see [6]):

$$2\kappa_w^f(\alpha) + \lambda(f(\alpha) - f^0(\alpha)) = 0, \quad \forall \alpha \in V. \tag{13}$$

Algorithms We use the linearized Gauss Jacobi method to find an approximate solution:

$$\begin{cases} f^{(0)} = f^0 \\ f^{(t+1)}(\alpha) = \dfrac{\lambda f^0(\alpha) + \sum_{\beta \sim \alpha} \gamma_w^{f^{(t)}}(\alpha, \beta) f^{(t)}(\beta)}{\lambda + \sum_{\beta \sim \alpha} \gamma_w^{f^{(t)}}(\alpha, \beta)}, \forall \alpha \in V, \end{cases} \tag{14}$$

where γ_w^f is defined in (7). Starting with the initial data f^0, each iteration of (14) relates the new value $f^{(t+1)}(\alpha)$ to $f^0(\alpha)$ and to a weighted average of the filtered data in the neighborhood of α. The contribution of f^0 is constant and dictated

by the scale parameter λ. As the decomposition of the successive residuals is performed, the scales are getting greater and the fidelity term has more impact on the decomposition.
For an unweighted graph, the algorithm (14) corresponds to the one proposed in [10] for the restoration of noisy images.

3.3 Digital Multiscale Hierarchical Decomposition on Graphs

In this Section, we propose a digital version of the multiscale hierarchical decomposition TNV, based on the framework introduced above. This digital version allows us to extend the TNV methodology to arbitrary graphs, including meshes and point clouds. The graph processing has another important advantage: it makes it possible to naturally consider nonlocal interactions between data [6] and thus enforces the nonlinearity of the decompositions.
Let us first review some aspects of the construction of weighted graphs. Data defined on graphs fall into two categories: organized and unorganized. For the former, the graph structure is known *a priori* while for the latter, a neighborhood graph should be considered. In this paper, we use the k-nearest neighbors graph where the nearest neighbors are selected according to a distance measure between vertices. Let V be a set of vertices, $E \subset V \times V$ a set of edges and f^0 a real function defined on V. The function f^0 is used to define a distance measure $d(\alpha, \beta)$ between each adjacent vertices α and β. For example, if f^0 is the function that assigns to each vertex its coordinates in the Euclidean space, we can use the Euclidean distance as a distance measure. If the graph represents an image domain and f^0 is the intensity function, one can consider the Euclidean distance between the intensity components for each pixel. If the data are organized, one can introduce feature vectors for each vertex and consider the distances between these feature vectors. A classical example is the introduction of patches around each pixel in the case of image denoising (see [6]). The graph now consists in of set of vertices, a set of edges and a distance measure d. A similarity measure w is defined, based on the distance d. w should be a non-increasing function that maps the range space of d to the interval $[0, 1]$. The ones we use in this paper are:
1) $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$
2) $w(\alpha, \beta) = e^{\frac{-d^2(\alpha,\beta)}{\sigma^2}}$
The estimation of σ is addressed later in this paper. In particular, we will consider local estimation of σ for each pixel in the case of image decomposition.
We have now a weighted graph $G = (V, E, w)$ and a function f^0 on V. We perform a first decomposition using equation (8) at an initial scale λ_0 and then iterate the process as in TNV, following the same geometric progression of scales.
The graph framework extends the TNV method. It has the following advantages, in contrast to the TNV approach:
1) The nonlocal processing is embedded in the graph structure.
2) It introduces adaptation within the graph weights.
3) It allows the decomposition of any data on graphs (meshes, curves, ...).

The decomposition equations are already in digital form consisting in a system of nonlinear equations (13) for each decomposition step. For a fixed index k, we end up with the same decomposition as in (2). Once general graph topologies and weights are considered, our approach generalizes and extends the TNV approach.

3.4 Parameter Selection

In this Section, we propose a method to select the parameters of the digital multiscale hierarchical decomposition. The parameters that we consider here are the initial scale λ_0 and σ, the kernel bandwidth in the case of an exponential weight. For images, we propose a local estimation of σ at each vertex.

Image Data For the nonlocal decomposition of images, we consider neighborhood graphs, mainly the k-nearest neighbors graph. In this case, we look for the k-nearest neighbors inside a window surrounding each pixel. This latter graph is coupled with a 4-adjacency grid graph, which will be denoted as k-NNG$_4$ in the sequel. For images we take $k = 25$ or $k = 10$. This choice allows for nonlocal processing while keeping the associated computational cost low. The edge weights $w(\alpha, \beta)$ play an important role in controlling the diffusion around each pixel. An efficient weight function should incorporate local informations embedded in the graph structure and the data. The exponential weight function is a good candidate for this. For each pixel, a 5×5 patch window is considered. More generally, for a $N \times N$ patch surrounding a pixel α, we take σ_α to be an estimation of the standard deviation of the intensity values in the patch. Such an approach amounts to consider the intensities as local independent random variables and to estimate their standard deviation. This standard deviation will serve to control the diffusivity of the regularization. We use an empirical estimation for the variance. If α is a pixel and B_α is the patch of size N_α surrounding α, then the estimate is:

$$\sigma_\alpha = \sqrt{\frac{1}{N_\alpha} \sum_{k \in B_\alpha} (f^0(k) - M_\alpha)^2} \ , \tag{15}$$

where M_α is the empirical mean computed inside the patch:

$$M_\alpha = \frac{1}{N_\alpha} \sum_{k \in B_\alpha} f^0(k) \ . \tag{16}$$

Then, $w(\alpha, \beta) = e^{\frac{-d^2(\alpha,\beta)}{\sigma_\alpha \sigma_\beta}}$. Other estimates can be used, in particular the median-based estimate used in [12].
For the parameter λ_0, a first approach could consist in using the analytical formula given in [10]. However, as the aim of the first minimization is to isolate the cartoon part, the parameter λ_0 can hardly be tied to an unconstrained minimization as in the denoising case. Our aim here is the decomposition of the image and we do this by considering a global standard deviation estimated as

in (15). This time we consider the entire image and let λ_0 be a multiple of $\frac{1}{\sigma}$. In our numerical experiments, we take $\lambda_0 = \frac{1}{2\sigma}$, so the texture below scale $\frac{1}{\lambda_0} = 2\sigma$ is unresolved in the first residual v_0. Here again more robust estimates can be used. For the case of color images, we use a single regularization for each color channel while the 1-Laplace operator is the same for all the components thus taking into account the correlation between the different channels.

Meshes In the case of meshes, the graph structure is known *a priori*. The function to regularize is the one that maps each vertex to its coordinates in the Euclidean space. The decomposition can be seen as a preprocessing task. Indeed, it has the advantage of separating different detail levels in a structure generally containing several thousands of points. To introduce adaptation in the decomposition, we use the weight function $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$. Here $d(\alpha, \beta)$ is the Euclidean distance between α and β.

Point Clouds The same methodology can be applied to unorganized sets of data. This time, the graph structure has to be constructed. When considering a neighborhood graph, one should check that the resultant graph is connected. If the k-nearest neighbors graph is choosen, the value of k has to fit this requirement.

4 Results

In this Section, we show some results of the proposed digital multiscale hierarchical decomposition on graphs. The approach is illustrated on grayscale and color images as well as on meshes and point clouds. The reader may consult the electronic version of the paper to accurately see the details in the provided decompositions.

4.1 Images

First, we illustrate the approach with a grayscale image. Figure 1 shows the 5 first levels of decomposition of an initial noisy finger image. Each reconstruction step is shown in a column in the second to fifth rows and the original image is in the first row. Level 1 corresponds to the first regularization u_0, level 2 to $u_0 + u_1$, until level 5, that shows $\sum_{i=0}^{4} = u_i$. Last column shows the last residual, in this case $v_4 + 128$. Results provided in the first row of Figure 1 correspond to the use of an unweighted 4-adjacency grid graph, i.e. $w(\alpha, \beta) = 1$ for all $(\alpha, \beta) \in E$ with λ_0 set to 0.02. One can see that this configuration produces blurred edges in the first steps of the decomposition. All the next rows correspond to results with our proposed extension. Third row presents results with a weighted 8-adjacency grid graph. The distance between two adjacent vertices is evaluated pixel-wise, an exponential weight is used with $\sigma = 40$, and λ_0 is set to 0.02. The use of a weighted graph enables an adaptive decomposition with less blur but much

more texture is present in the first levels. Fourth row presents results with a weighted 25−NNG_4. The distance between two adjacent pixels is evaluated by introducing 5×5 patches around each pixel. The weight function is exponential with σ evaluated globally and $\lambda_0 = \frac{1}{2\sigma}$. This time with nonlocal weights, few texture is preserved in the first levels and the edges of the structure are sharper. One can also see that the noise part in the initial image has not been obtained at the last level of the decomposition and can be found in the residual. The last row of Figure 1 uses the same graph structure as in the former one. The same exponential weight is used but this time, σ is local to each pixel and evaluated as detailed in Section 3.4 as well as for λ_0. We see that in this latter case, the texture has completely disappeared from the first level and only the geometric part has been captured. Also, the reconstruction is more homogeneous, and each decomposition brings a distinct part of the texture, making the scale separation more explicit. This illustrates how a careful selection of the parameters (initial λ_0 and graph weights) is important to obtain an accurate nonlocal decomposition.

Figure 2 shows the results of the decomposition of a textured image. The initial image (first row) contains 5 different textures, each one with a different direction. The purpose here is to isolate the different texture regions across the levels. We also would like to verify that the coarser scale textures are reconstructed first. Second row presents a 5-step decomposition using a 4-adjacency grid graph with unweighted edges and $\lambda_0 = 0.005$. First level fails to separate the different textured regions, but some scale separation properties start appearing in level 3. Third row presents results obtained by considering a 10-NNG_4, with weighted edges (exponential), σ is computed locally as in Section 3.4 inside a 5×5 patch, and λ_0 is selected as in Section 3.4. In this example, the first level succeeds to capture an important aspect of the geometry. The reconstruction is faster and the scale separation is more explicit than in the previous row. Finally, the same comparison is performed on a color image in Figure 3. The decomposition has been considered for each color channel, with the same 1-Laplace operator for all the channels. Results in the first row are obtained with $\lambda_0 = 0.05$ on an unweighted 4-adjacency grid graph. Second row presents a nonlocal decomposition with parameters automatically estimated (inside a 5×5 patch for σ_α). Here again, the first level of the nonlocal adaptive decomposition succeeds in capturing the geometrical part of the initial image. Due to the variety of textures present in the initial image, the reconstruction in the case of the nonlocal decomposition is slower and one should look at finer scales in order to completely recover the initial image. Indeed, with our nonlocal approach, repetitive structures (such as the window patterns) are considered as texture and removed from the first level that contains only the sole structure part of the image.

4.2 Meshes and Point Clouds

Here we provide numerical experiments illustrating the application of the multiscale hierarchical decomposition on graphs to meshes and point clouds. For

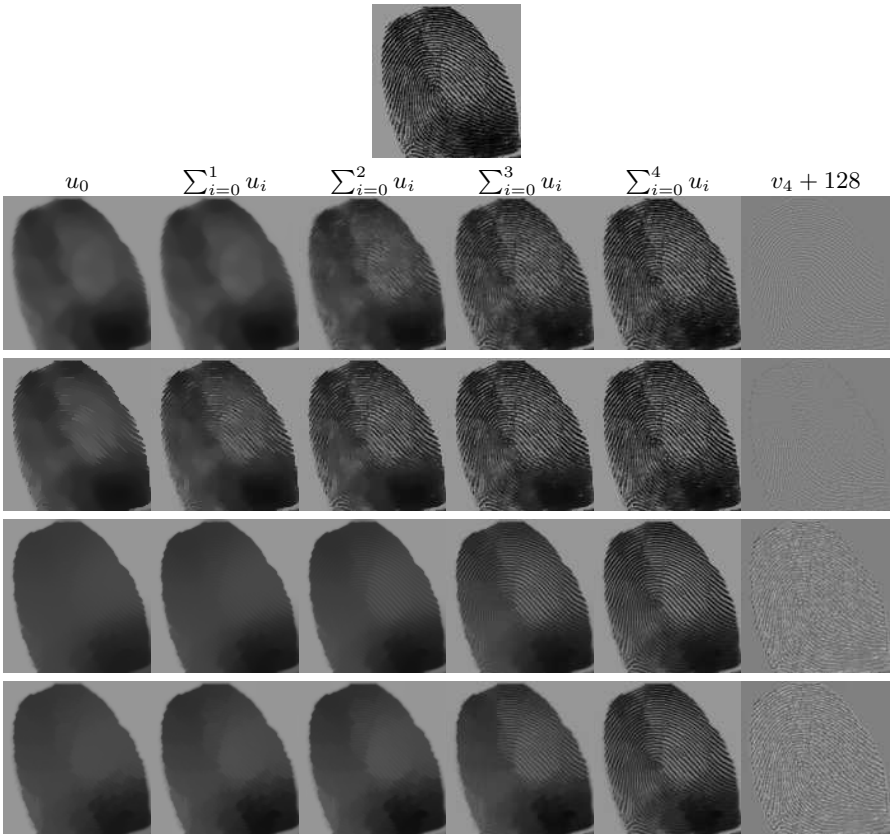

Fig. 1. Decomposition of a noisy finger image. Fist row: original image. Second row: 4-adjacency unweighted grid graph. Third row: 8-adjacency grid graph with exponential weight. Fourth row: 25-NNG_4 with exponential weight and 5×5 patches. Fifth row: 25-NNG_4 with exponential weight and 5×5 patches, σ_α is evaluated at each pixel, λ_0 is evaluated automatically. See text for more details.

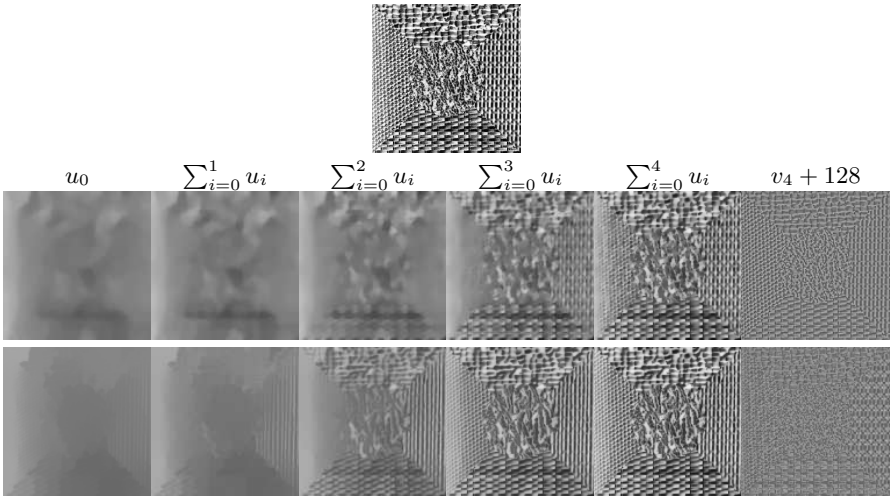

Fig. 2. Decomposition of a textured image. Fist row: initial image. Second row: 4-adjacency grid graph, unweighted. Third row: 10-NNG$_4$, exponential weight, σ_u is evaluated at each pixel, λ_0 is evaluated automatically. See text for more details.

meshes, the graph structure is given *a priori*. Figure 4 shows the results of a 10-step decomposition performed on a 3-dimensional hand mesh. The parameter λ_0 was set to 0.1, the distance between two vertices is the Euclidean distance, and the edges are weighted with: $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$. Rows 1 and 2 show two different views (interior and exterior) of the same hand mesh. The decomposition performs well on the first level by simplifying the mesh structure. Different details are recovered as the decompositions are performed. Finally, in Figure 5, we illustrate the denoising capability of the digital decomposition by applying it to a noisy toroidal helix. A 20-nearest neighbors graph has been considered, λ_0 was set to 0.1, the distance between two vertices is the Euclidean distance and the edges are weighted with: $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$ where σ was set to the maximum of all the distances between two connected vertices. The first decomposition has a strong simplification impact and reveals the structure of the helix. A denoised toroidal helix is recovered as the successive decompositions are performed.

5 Conclusion

In this paper, we generalized the hierarchical multiscale decomposition of Tadmor, Nezzar and Vese [1] to general data defined on graphs. The decomposition is based on a discrete variational framework. For images, the introducton of nonlocal interactions enables a finer decomposition. Moreover, since the proposed formulation considers arbitrary graphs, unusual domains such as meshes point clouds can be considered. Finally, we also have addressed the crucial problem of parameter selection to have a fully automatic decomposition.

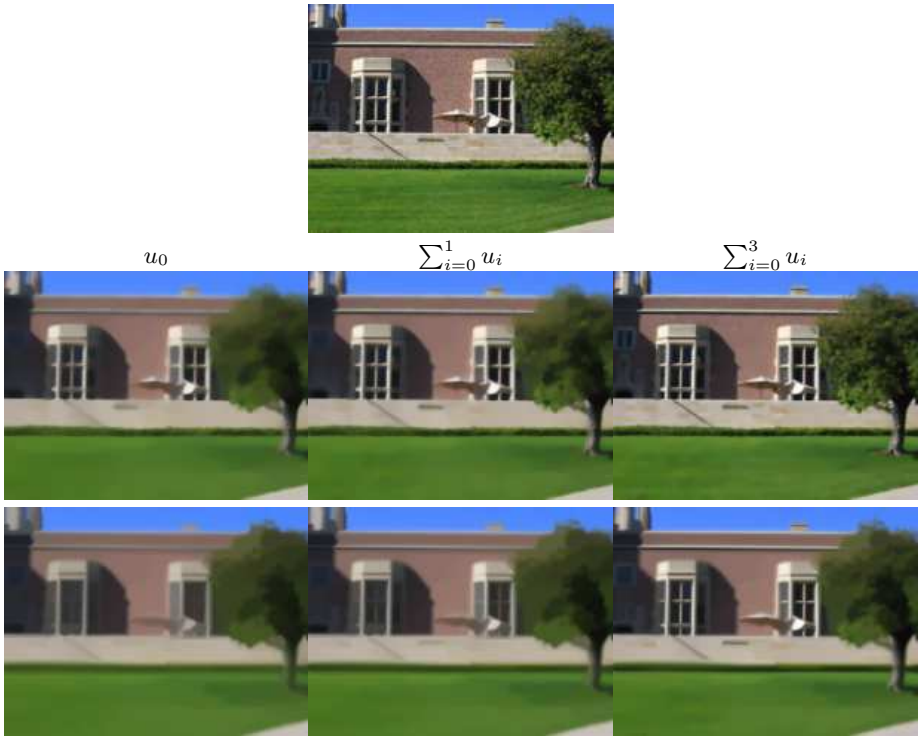

Fig. 3. Decomposition of a color image. Fist row: initial image. Second row: 4-adjacency grid graph, unweighted. Third row: 10-NNG$_4$, exponential weight, σ_α is evaluated at each pixel, and λ_0 is evaluated automatically.

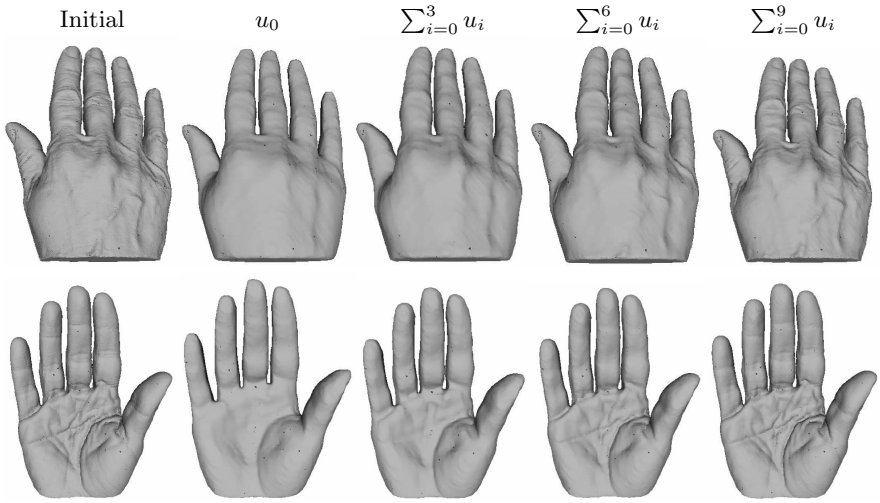

Fig. 4. Decomposition of a hand mesh. Initial scale $\lambda_0 = 0.1$. See text for more details.

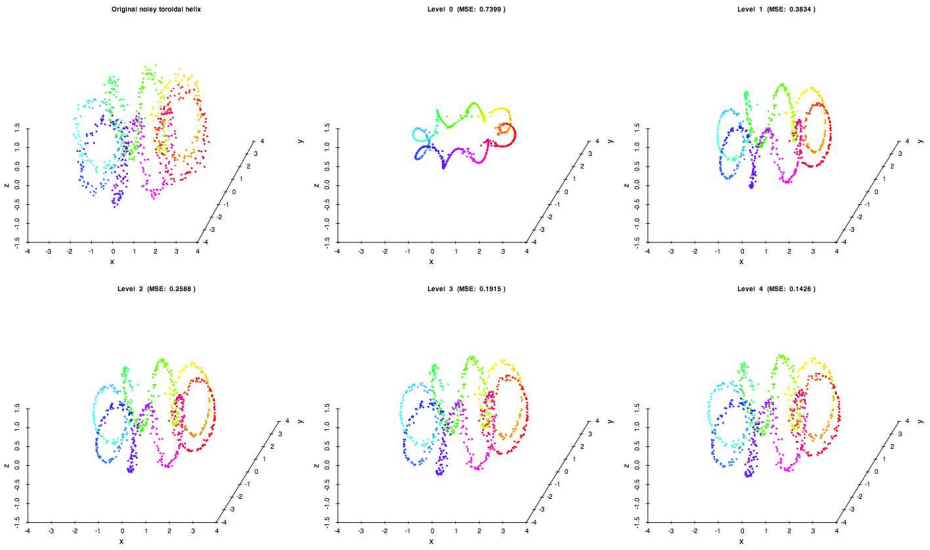

Fig. 5. Decomposition of a noisy toroidal helix. Initial scale $\lambda_0 = 0.1$, 20-nearest neighbors graph. See text for more details.

References

1. Tadmor, E., Nezzar, S., Vese, L.: A multiscale image representation using hierarchical (BV, L2) decompositions. Multiscale Modeling and Simulation **2** (2004) 554–579
2. Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. University Lecture Series. American Mathematical Society, Boston, MA, USA (2001)
3. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60** (1992) 259–268
4. Buades, A., Coll, B., Morel, J.M.: Nonlocal image and movie denoising. International Journal of Computer Vision **76** (2008) 123–139
5. Buades, A., Coll, B., Morel, J.M.: Image denoising methods. A new non-local principle. Siam Review **52** (2010) 113–147
6. Elmoataz, A., Lézoray, O., Bougleux, S.: Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. IEEE Transactions on Image Processing **17** (2008) 1047–1060
7. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. Multiscale Modeling and Simulation **7** (2008) 1005–1028
8. Ohtake, Y., Belyaev, A., Seidel, H.P.: A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In: Proceedings of the Shape Modeling International. (2003) 153
9. Tadmor, E., Nezzar, S., Vese, L.: Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation. Communications in Mathematical Sciences **6** (2008) 281–307

10. Chan, T.F., Osher, S., Shen, J.: The digital TV filter and nonlinear denoising. IEEE Transactions on Image Processing **10** (2001) 231 – 241
11. Lezoray, O., Elmoataz, A., Bougleux, S.: Graph regularization for color image processing. Computer Vision and Image Understanding **107** (2007) 38–55
12. Kervrann, C.: An adaptive window approach for image smoothing and structures preserving. In: Proceedings of European Conference on Computer Vision. (2004)