Single-Image Super-Resolution via Linear Mapping of Interpolated Self-Examples

Marco Bevilacqua, Aline Roumy, Christine Guillemot, Alberi Morel

Single-image super-resolution via linear mapping of interpolated self-examples

Marco Bevilacqua, Aline Roumy, *Member, IEEE,* Christine Guillemot, *Fellow, IEEE,* and Marie-Line Alberi Morel

Abstract

This paper presents a novel example-based single-image super-resolution (SR) procedure, that upscales to high-resolution (HR) a given low-resolution (LR) input image without relying on an external dictionary of image examples. The dictionary instead is built from the LR input image itself, by generating a "double pyramid" of recursively scaled, and subsequently interpolated, images, from which self-examples are extracted. The upscaling procedure is multi-pass, i.e. the output image is constructed by means of gradual increases, and consists in learning special linear mapping functions on this double pyramid, as many as the number of patches in the current image to upscale. More precisely, for each LR patch, similar self-examples are found, and, thanks to them, a linear function is learned to directly map it into its HR version. Iterative back projection is also employed to ensure consistency at each pass of the procedure. Extensive experiments and comparisons with other state-of-the-art methods, based both on external and internal dictionaries, show that our algorithm can produce visually pleasant upscalings, with sharp edges and well reconstructed details. Moreover, when considering objective metrics like PSNR and SSIM, our method turns out to give the best performance.

Index Terms

super resolution, example-based, regression, neighbor embedding

I. Introduction

Super-resolution (SR) refers to a family of techniques that aim at increasing the resolution of given images. Nowadays, many applications, e.g. video surveillance and remote sensing, require the display of images at a considerable resolution, that may not be easy to obtain given the limitations of physical imaging systems or

This work was supported by the joint research lab INRIA-Alcatel Lucent Bell Labs.

M. Bevilacqua is with IMT Institute for Advanced Studies Lucca, 55100 Italy (e-mail: marco.bevilacqua@imtlucca.it). This work was performed while he was Ph.D. student at the SIROCCO research team in INRIA.

A. Roumy and C. Guillemot are with the SIROCCO research team in INRIA, Institut national de recherche en informatique et en automatique, Rennes, 35042 France (e-mail: firstname.lastname@inria.fr).

M-L. Alberi Morel is with Alcatel-Lucent Bell Labs France, Centre de Villarceaux, 91620 France (e-mail: marie line.alberi-morel@alcatel-lucent.com).

environmental conditions. Moreover, with the spread of digital technologies, there is a tremendous amount of user-produced images collected in the years, that are valuable but may be affected by a poor quality. Therefore, techniques to augment the resolution of an image, i.e. the total number of pixels, and contextually improve its visual quality, are particularly appealing.

SR methods are traditionally categorized according to the number of input images: when several low-resolution (LR) input images are available we speak about *multi-frame* SR algorithm; vice versa, when the LR input image is unique, we have the *single-image* SR problem. In both cases, as an output, a unique high-resolution (HR) super-resolved image is produced.

Multi-frame SR methods, of which a good overview is given in [1], are the first ones that have been studied, since the SR problem first appeared in the scientific community [2]. Here, the multiple LR input images are considered as different views of the same scene, taken with sub-pixel misalignment, i.e. each image is seen as a degraded version of an underlying HR image to be estimated, where the degradation processes can include blurring, geometrical transformations, and down-sampling. Single-image SR, instead, aims at constructing the HR output image from as little as a single LR input image. The problem stated is an inherently ill-posed problem, as there can be several HR images generating the same LR image. Single-image SR is deeply connected with traditional "analytical" interpolation, since they share the same goal. Traditional interpolation methods, e.g. bicubic interpolation, by computing the missing pixels in the HR grid as averages of known pixels, implicitly impose a "smoothness" prior. However, natural images often present strong discontinuities, such as edges and corners, and thus the smoothness prior results in producing ringing and blurring artifacts in the output image. Single-image SR algorithms can be broadly classified into two main approaches: *interpolation-based methods* [3]–[5], which, possibly in a nonparametric fashion, follow the interpolation approach by posing more sophisticated statistical priors; and *machine learning (ML) based* methods.

In particular, the latter, by taking advantage of the powerfulness of ML techniques, have shown to give very challenging results, and many algorithms appeared in the literature in the recent years. ML-based algorithms can consist in pixel-based procedures, where each value in the HR output image is singularly inferred via statistical learning [6], [7], or patch-based procedures, where the HR estimation is performed thanks to a dictionary of correspondences of LR and HR patches (i.e. squared blocks of image pixels). ML-based SR that makes use of patches is also referred to as *example-based* SR [8]. In the upscaling procedure the LR input image itself is divided into patches, and for each LR input patch a single HR output patch is reconstructed, by observing the "examples" contained in the dictionary.

Example-based methods mainly vary in two aspects: the patch reconstruction method used and the typology of dictionary. As for the method to perform the single patch reconstructions, we have again two main categories: coding-based reconstruction methods and direct mapping (DM). Neighbor embedding (NE) belongs to coding-based methods. In NE-based SR [9]–[11], for each LR input patch we select K similar LR examples in the dictionary by nearest neighbor search (NNS), and a linear combination of these neighbors is computed to possibly approximate the input patch. The corresponding HR neighbors are then similarly combined, i.e. by using the same weights

computed with the LR patches, to generate the HR output patch. This approach relies on what in [12] is called "manifold assumption": the linear combination weights are meant to capture the local geometry of a manifold on which the LR patches are supposed to lay; by applying the same weights to reconstruct the HR patches, we implicitly assume that they too lay on a manifold with similar local structures. In order to enforce the assumption of manifold similarity between the two distinct spaces represented by the LR and HR patches, Gao *et al.* propose in [13] to compute the weights in a subspace common to the LR and HR patches. The method therefore assumes the existence of a common low-dimensional space that preserve some meaningful characteristics of the patches.

Example-based SR via sparse representations [14], [15] also falls within the family of coding-based methods, and can be considered very close to NE-based SR: however, here, the weights are not computed on neighbors found by NNS, but typical sparse coding algorithms are employed. The method presented in [16] bridges the two approaches (NE and sparse representations), by proposing a "sparse neighbor embedding" algorithm. In [17], instead, another sparse-representation-based SR algorithm is presented, which, in addition to a sparse example-based term, uses several regularization terms to globally optimize the image generation process.

All the methods so far require manifold similarity between the LR and HR spaces. A different patch reconstruction approach that does not rely on this assumption is given instead by direct mapping (DM). Example-based SR via DM [18]–[20] aims at finding, in fact, for each patch reconstruction, a function that directly map a given LR patch into its HR version. The mapping function is commonly found with traditional regression methods.

As for the second discriminating aspect of example-based SR, the typology of the dictionary, we have mainly two choices: an external dictionary, built from a set of external training images, and an "internal" one, built without using any other image than the LR input image itself. This latter case exploits the so called "self-similarity" property, typical of natural images, according to which image structures tend to repeat within and across different image scales: therefore, patch correspondences can be found in the input image itself and possibly scaled versions of it. To learn these patch correspondences, that specifically take the name of *self-examples*, we can have one-step schemes [18], [21], [22] or schemes based on a pyramid of recursively scaled images starting from the LR input image [23]–[25]. Clearly, the advantage of having an external dictionary instead of an internal one lies in the fact that it is built in advance, while the internal one is generated online and updated at each run of the algorithm. However, external dictionaries have a considerable drawback: they are fixed and thus non-adapted to the input image. The study conducted in [26] also confirms the benefit of using internal statistics in patch-based image processing algorithms.

A. Main contributions

In this paper we present a novel example-based SR algorithm that makes use of an internal dictionary. The contributions are twofold. First, we build a "double pyramid", where the traditional image pyramid of [23] is juxtaposed with a pyramid of interpolated images. Second, HR patches are reconstructed through DM, whereas in [23] the HR patches are reconstructed via NE. Therefore, in the proposed algorithm, a linear mapping is learned on the double pyramid from the LR interpolated patches to the HR patches, and then applied to each interpolated LR

input patch. As said before, unlike NE, DM does not rely on any theoretical assumption (i.e. a manifold similarity between the LR and HR spaces), and is therefore preferable.

The rationale for the interpolation operation is to make the computation of the single mapping functions via regression more robust (LR and HR patches, in fact, turn out to have the same sizes). Moreover, a Tikhonov regularization is added to the problem to provide numerical stability in the computation, since the linear mapping requires a matrix inversion. A *double-pyramid-like* scheme has also been introduced in [22]. However, our proposed algorithm differs from [22] in two ways. First, the reconstruction in [22] is based on a neighbor embedding technique analyzed in [27] (NE with a sum-to-one constraint). This NE technique was shown to have a performance highly dependent on the number of neighbors chosen. More precisely, when this number is equal to the size of the LR input vectors, the performance of the algorithms dramatically drops. As a second discriminating aspect, while [22] initializes the pyramid by down-scaling the LR input image only once, we instead initialize the pyramid with many sub-levels by recursively down-scaling the LR input image. We show that many sub levels are needed in order to better reconstruct a HR image. More precisely, we have that at the first iteration of the algorithm around 35% of the selected patches come from the pyramid levels below the first one (see Fig. 3 for details). The patches initially selected play a crucial role: in fact, since the whole algorithm is by nature recursive, it is very sensitive to its initialization.

All the listed contributions are possible at the cost of a slight increase in the complexity of the algorithm (i.e. +10.7% for a scale factor of 3 and +8.1% for a scale factor of 4; see Section IV-B for details), while having a beneficial effect on the quality performance, both in terms of objective metrics (PSNR and SSIM) and visual results.

B. Organization of the paper

The rest of the paper is organized as follows. Section II reviews the fundamentals of example-based SR using an internal dictionary, by giving also a general introduction of NE and DM methods. Then, in Section III, our algorithm is described in detail, by explaining how the internal dictionary is trained and the whole upscaling procedure. Before drawing the conclusions, Section IV presents some extensive experiments done: the different implementation choices are here validated and the algorithm is compared with other state-of-the-art methods, by showing visual and quantitative results.

II. Example-based SR with self-examples

A. Principles and notations

Single-image SR is the problem of estimating an underlying HR image, given only one observed LR image. The generation process of the LR image from the original HR image, that we consider, can be written as

$$I_L = (I_H * B) \downarrow_s , \tag{1}$$

where I_L and I_H are respectively the LR and HR image, B is a blur kernel the original image is convoluted with, which is typically modeled as a Gaussian blur [28], and the expression $\downarrow_s$ denotes a downsampling operation by a scale factor of s. The LR image in then a blurred and down-sampled version of the original HR image.

Example-based single-image SR aims at reversing the image generation model (1), by means of a dictionary of image *examples* that map locally the relation between an HR image and its LR counterpart, the latter obtained with the model (1). For general upscaling purposes, the examples used are typically in the form of patches, i.e. squared blocks of pixels (e.g. 3×3 or 5×5 blocks). The dictionary is then a collection of patches, which, two by two, form pairs. A pair specifically consists of a LR patch and its HR version with enriched high frequency details.

Example-based SR algorithms comprise two phases:

1) A *training phase*, where the above-mentioned dictionary of patches is built;
2) The proper *super-resolution phase*, that uses the dictionary created to upscale the input image.

As for the training phase, in the next section we discuss how to build an "internal" dictionary of patches, i.e. starting from as little as the LR input image. As for the SR phase, instead, in example-based algorithms the patch is also the reconstruction unit used in the upscaling procedure. In fact, the LR input image is partitioned into patches; for each single LR input patch, then, by using the LR-HR patch correspondences in the dictionary, a HR output patch is constructed. The HR output image is finally built by re-assembling all the reconstructed HR patches. In Section II-C we revise the two main patch reconstruction approaches: neighbor embedding and direct mapping.

Hereinafter in this paper we will use the following notation to indicate the different kinds of patches. $\mathcal{X}^l = \{\mathbf{x}_i^l\}_{i=1}^{N_x}$ will denote the set of LR patches into which the LR input image is partitioned; similarly, $\mathcal{X}^h = \{\mathbf{x}_i^h\}_{i=1}^{N_x}$ will denote the set of reconstructed HR patches, that will form the HR output image. Each patch is expressed in vector form, i.e. its pixels are concatenated to form a unique vector. As for the dictionary, $\mathcal{Y}^l = \{\mathbf{y}_i^l\}_{i=1}^{N_y}$ and $\mathcal{Y}^h = \{\mathbf{y}_i^h\}_{i=1}^{N_y}$ will be respectively the sets of LR and HR example patches. In each patch reconstruction, then, the goal is to predict an HR output patch $\mathbf{x}_i^h$, given the related LR input patch $\mathbf{x}_i^l$, and the two coupled sets of patches, $\mathcal{Y}^l$ and $\mathcal{Y}^h$, that form the dictionary. Fig. 1 shows in a simple manner the operating diagram of an example-based SR algorithm, where the input image is partitioned into LR patches, a dictionary of patch correspondences is exploited, and new HR patches are generated and subsequently re-assembled to create the super-resolved output image.

B. Searching for self-examples

In example-based SR, when performing the training phase, we speak about an internal learning when, instead of making use of external training images, we derive the patches directly from the input image and processed versions of it. Local image structures, that can be captured in the form of patches, tend to recur across different scales of an image, especially for small scale factors. We can then use the input image itself, conveniently up-sampled or down-sampled into one or several differently re-scaled versions, and use pairs of these images to learn correspondences of LR and HR patches, that will constitute our internal dictionary. We call the patches learned in this way *self-examples*. In this respect, there are two main kinds of learning schemes described in the literature:

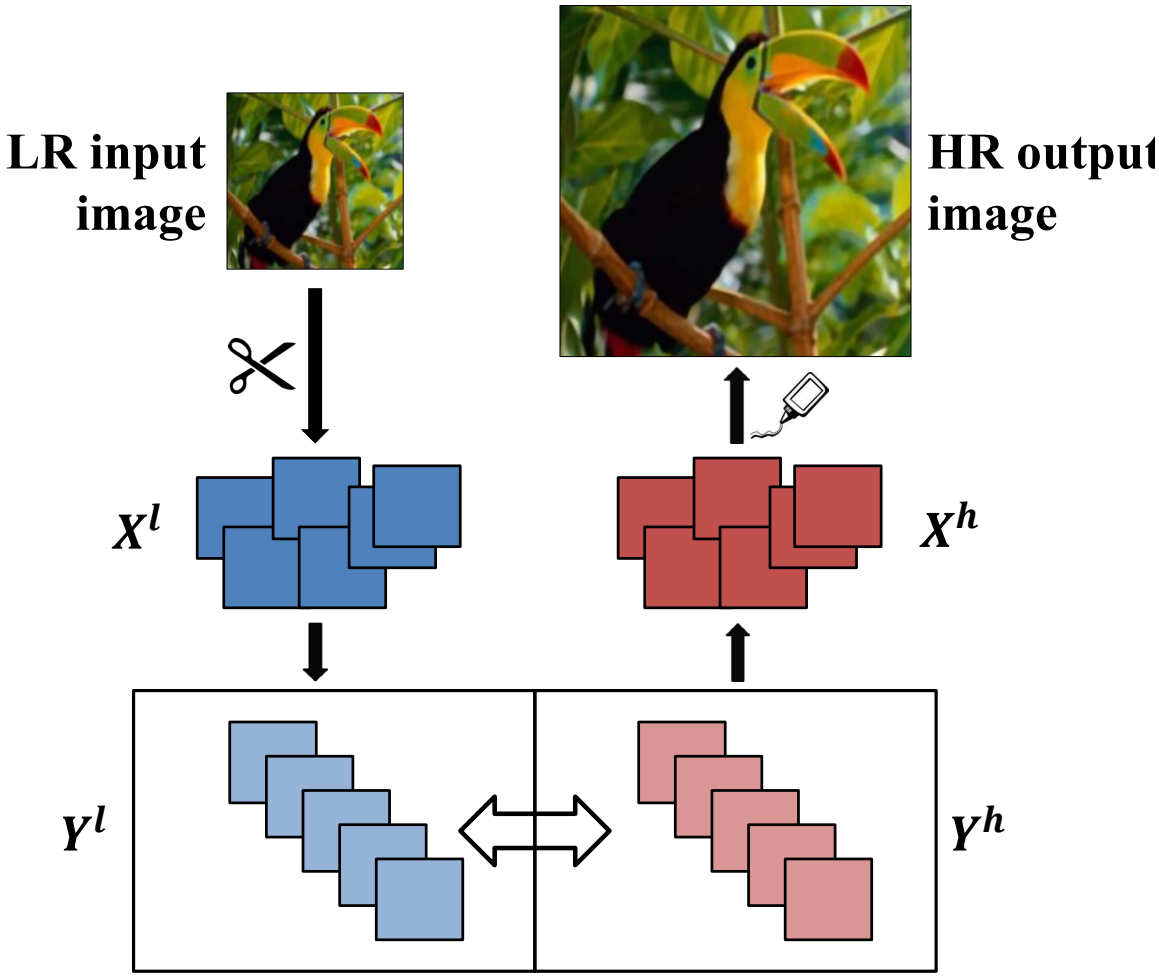


Fig. 1. Operating diagram of the example-based SR procedure.

one-step schemes like in [18], [21], [22], and schemes involving the construction of a pyramid of recursively scaled images [23]–[25].

One-step schemes are meant to reduce as much as possible the size of the internal dictionary, i.e. the number of self-examples to be examined at each patch reconstruction, and thus the computational time of the SR algorithm. In fact, from the input image only one pair of training images is constructed and thus taken into account for the construction of the dictionary. This approach is motivated by the fact that the most relevant patches correspondences can be found when the rescale factor employed is rather small. Only one rescaling is then sufficient to obtain a good amount of self-examples. Let $\mathcal{D}$ denote an image downscaling operator, s.t. $\mathcal{D}(I) = (I)\downarrow_p$, where p is a conveniently chosen small scale factor; and let $\mathcal{U}$ denote the dual upscaling operator, s.t. $\mathcal{U}(I) = (I)\uparrow_p$. Let I_L still indicate the LR input image. In [22], for example, $J_L = \mathcal{U}(\mathcal{D}(I_L))$, which represents a low-pass filtered version of the LR input image I_L, is used as source for the LR patches, whereas the HR example patches are directly sampled from the input image ($J_H = I_L$). Freedman and Fattal propose in [21] an equivalent approach, except that

a high-pass version of I_L ($J_H = I_L - J_L$) is used to get HR training patch. In [18], instead, we have $J_L = (I_L * B)$ and $J_H = \mathcal{U}(I_L)$: the LR examples patches are taken again from a low-pass filtered version of I_L, obtained by blurring the original image with a Gaussian blur kernel B, and the corresponding HR patches are taken from an upscaled version of I_L, which does not alter its frequency spectrum content.

The method described in [23] paved the way to several SR algorithms (e.g. [24], [25]), based on self-examples derived in an image pyramid. Differently from one-step learning methods, here several training images are constructed by recursively down-scaling the LR input image, thus forming a sort of overturned pyramid. Given a LR input patch, all the levels of this pyramid can be used to find similar self-examples, usually by performing a nearest neighbor search (NNS) (see Fig. 2).

To test the effective usefulness of constructing a full pyramid of images, we built a pyramid with 6 sub-levels as in [23] (the original LR image is recursively down-scaled 6 times). For each LR input patch, then, the $K = 9$ most similar self-examples (the K nearest neighbors) have been searched throughout the whole pyramid, and the histogram of all the selected neighbors has been drawn, where the histogram classes are the 6 sub-levels of the pyramid. Fig. 3 presents the histograms of the selected neighbors for two different images.

As we can observe from Fig. 3, it is clear that the first sub-level, i.e. the image obtained with only one rescaling operation, is the most relevant source of self-examples. Nevertheless, in both cases nearly 35 percent of the neighbors still come from the other sub-levels, which is not a negligible percentage. Pyramidal scheme like in [23] are then preferable than one-step scheme like [18], [21], [22].

C. Patch reconstruction methods based on local learning

Once the dictionary of self-examples is built, i.e. we have the two dictionary sets $\mathcal{Y}^l$ and $\mathcal{Y}^h$ containing, respectively, LR and HR example patches, the proper SR upscaling procedure is ready to start. In this respect, example-based SR algorithms consist in patch-based procedures, where the HR output image is built by means of single patch reconstructions, as many as the number of LR patches the LR input image is partitioned into.

For example-based SR, two main approaches to reconstruction are possible: neighbor embedding (NE) and direct mapping (DM). In both cases, for each input patch $\mathbf{x}_i^l$, LR and HR *local* training sets are formed, by performing a nearest neighbor search (NNS). A desired number of neighbors of the LR input patch $\mathbf{x}_i^l$ is searched among the LR self-examples $\mathcal{Y}^l$, and consequently an equal number of HR neighbors is determined. Let Y_i^l indicate the matrix collecting, column by column, the selected LR neighbors, and let Y_i^h indicate the matrix of the corresponding HR neighbors. Thanks to the local sets Y_i^l and Y_i^h, the unknown HR patch $\mathbf{x}_i^h$ is then predicted. The whole *local learning* procedure can be summarized in 3 steps.

1) **Nearest Neighbor Search (NNS)**: The local training sets Y_i^l and Y_i^h are determined.
2) **Model generation**: A model $\mathcal{M}_i$ for the local reconstruction is computed. $\mathcal{M}_i$ generally depends on both the LR input patch and the local training sets: $\mathcal{M}_i = \mathcal{M}(\mathbf{x}_i^l, Y_i^l, Y_i^h)$.
3) **Prediction**: The model $\mathcal{M}_i$ is applied to actually predict the HR output patch.

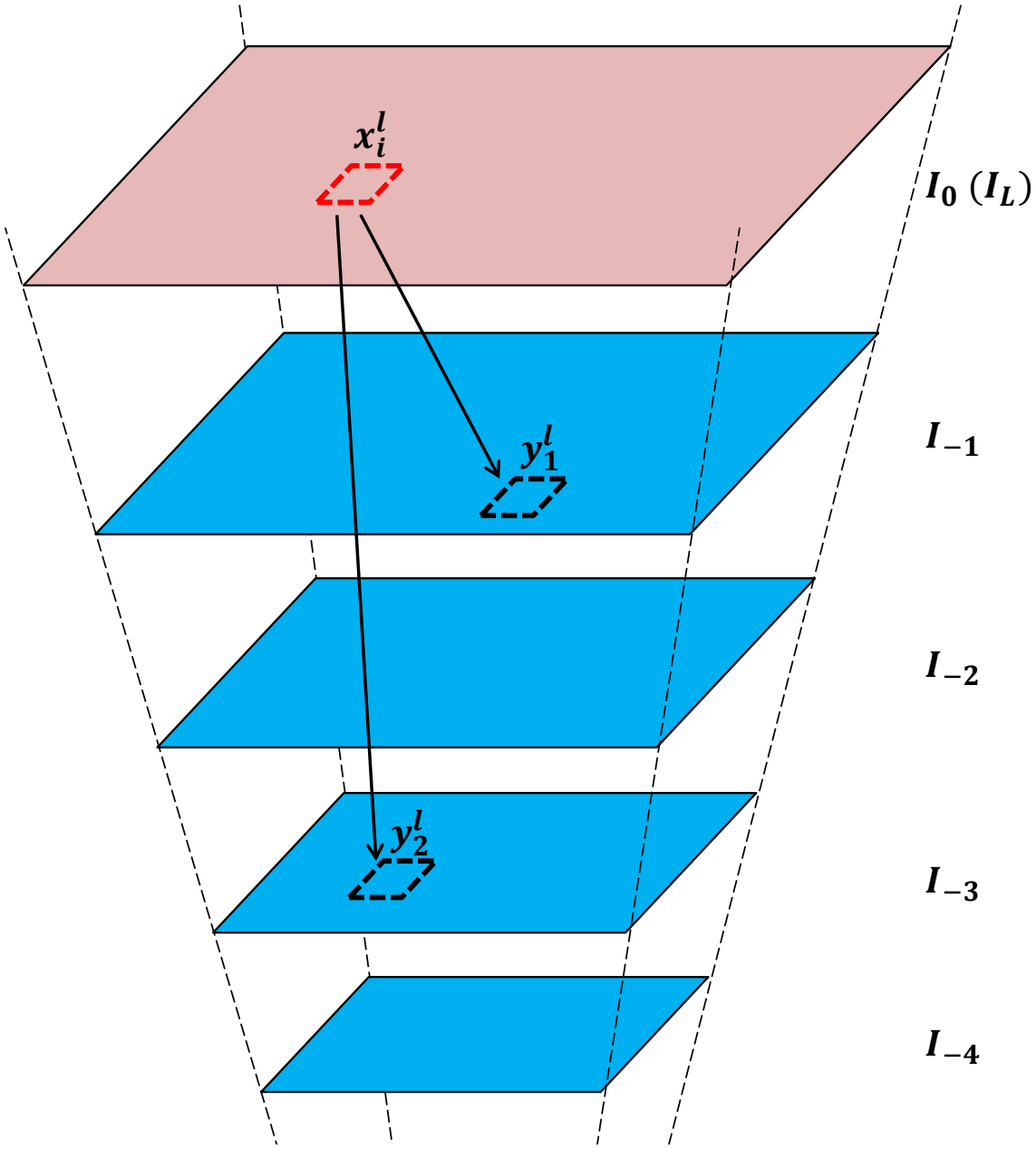


Fig. 2. Pyramid of recursively scaled images (in this case with 4 sub-levels), where the top level (I_0) is represented by the LR input image I_L. Given a LR input patch $\mathbf{x}_i^l$, a desired number of "neighbors" ($\mathbf{y}_1^l, \mathbf{y}_2^l, \dots$) can be found at any level of the pyramid.

NE and DM differ in steps 2 and 3 of the local learning procedure above. In NE, the reconstruction model $\mathcal{M}_i$ consists of a vector of weights $\mathbf{w}_i \in \mathcal{R}^K$ (where K is the number of neighbors chosen), that identifies a linear combination of the LR neighbors Y_i^l. The weights are computed w.r.t. the LR input patch and its neighbors ($\mathbf{w}_i = \mathcal{M}(\mathbf{x}_i^l, Y_i^l)$). In [23], e.g., the single weight $w_i(j)$, related to the neighbor $\mathbf{y}_j^l$ found in the pyramid, is an exponential function of the distance between the latter and the LR input patch, according to the non-local means

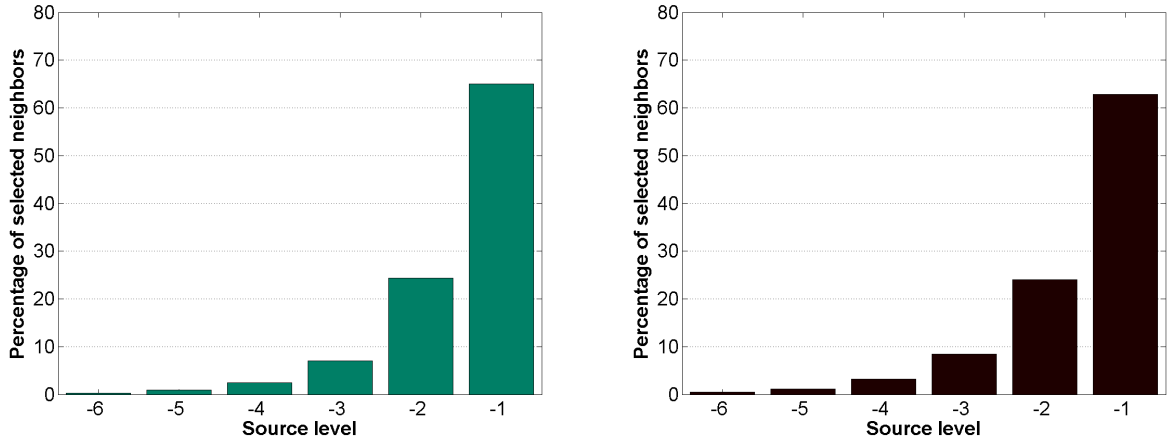


Fig. 3. Percentage of selected neighbors for each sub-level of the pyramid, for two different images.

(NLM) model [29].

$$w_i(j) = \frac{1}{C} e^{-\frac{\|\mathbf{x}_i^l - \mathbf{y}_j^l\|_2^2}{t}} , \tag{2}$$

where t is a parameter to control the decaying speed and C is a normalizing constant to make the weights sum up to one.

In other NE methods, instead, the weights are meant to describe a linear combination that approximates the LR input patch (i.e. $\mathbf{x}_i^l \approx Y_i^l \mathbf{w}_i$). In [9]–[11], e.g., the weights are computed as the result of a least squares problem with a sum-to-one constraint:

$$\mathbf{w}_i = \arg\min_{\mathbf{w}} \|\mathbf{x}_i^l - Y_i^l \mathbf{w}\|^2 \quad \text{s.t.} \quad \mathbf{w}^T \mathbf{1} = 1 . \tag{3}$$

The formula (3) recalls the method used in Locally Linear Embedding (LLE) [30] to describe a high-dimensional point lying on a manifold through its neighbors. In [27], instead, the sum-to-one constraint is replaced by a nonnegative condition, thus leading to nonnegative weights.

$$\mathbf{w}_i = \arg\min_{\mathbf{w}} \|\mathbf{x}_i^l - Y_i^l \mathbf{w}\|^2 \quad \text{s.t.} \quad \mathbf{w} \geq 0 . \tag{4}$$

In [13], finally, the weight computation is performed in an appropriate subspace: the LR input patch and its HR neighbors are expressed as lower-dimensional vectors, thanks to convenient projection matrices. The weights are then found by minimizing the error in the new subspace with no explicit constraint:

$$\mathbf{w}_i = \arg\min_{\mathbf{w}} \|P_i^l \mathbf{x}_i^l - P_i^h Y_i^h \mathbf{w}\|^2 . \tag{5}$$

Once the vector of weights $\mathbf{w}_i$ is computed, the prediction step (Step 3) of the NE learning procedure consists in generating the HR output patch $\mathbf{x}_i^h$ as a linear combination of the HR neighbors Y_i^l, by using the same weights:

$$\mathbf{x}_i^h \approx Y_i^h \mathbf{w}_i . \tag{6}$$

Fig. 4a depicts the scheme of the patch reconstruction procedure in the NE case. The model, i.e. the weights, is totally learned in the LR space, and then applied to the HR local training set to generate the HR output patch.

In direct mapping (DM) methods [18]–[20], instead, the model is a regression function f_i that directly maps the LR input patch into the HR output patch. The function is learned by taking into account the two local training sets ($f_i = \mathcal{M}(Y_i^l, Y_i^h)$), by minimizing the empirical fitting error between all the pairs of examples. An appropriate regularization term is placed to make the problem well-posed. We then have:

$$f_i = \arg\min_{f \in \mathcal{H}} \sum_{j=1}^{K} \|\mathbf{y}_j^h - f(\mathbf{y}_j^l)\|_2^2 + \lambda \|f\|_{\mathcal{H}}^2 \,, \tag{7}$$

where $\mathcal{H}$ is a desired Hilbert function space and $\lambda \geq 0$ a regularization parameter. Examples similar to $\mathbf{x}_i^l$ and their corresponding HR versions are then used to learn a unique mapping function, which is afterwards simply applied to $\mathbf{x}_i^l$ to predict the HR output patch (Step 3):

$$\mathbf{x}_i^h = f_i(\mathbf{x}_i^l) \tag{8}$$

Fig. 4b depicts the scheme of the patch reconstruction procedure also in the DM case.

An advantage of DM w.r.t. NE is that it can enable fast procedures, while presenting only slightly degraded performance, as done in [31]. By allowing a mismatch in the neighborhood computation (neighborhood of the closest atom in the dictionary rather than neighborhood of the input patch), the mappings can be pre-computed and stored. Hence the fast implementation.

III. Proposed algorithm

In this section we present our novel SR algorithm, based on an internal dictionary of self-examples. Starting from the image pyramid described in Section II-B, we propose a modified scheme with a "double pyramid", presented in Section III-A. The self-examples found in this scheme are used to gradually upscale the LR input image up to the final super-resolved image, according to the cross-level scale factor chosen for the pyramid. The upscaling procedure employed falls within the local learning based reconstruction methods described in Section II-C. In particular, it is a direct mapping method, where each LR patch is mapped into its HR version by means of a specifically learned linear function. The whole upscaling procedure and a summary of the whole algorithm are given in Section III-B.

A. Building the double pyramid

The goal of our SR algorithm is to retrieve the underlying HR image I_H from a degraded LR version of its I_L, which is supposed to be originated according to the image generation model (1). We choose the blur kernel B to be a Gaussian kernel with a given variance σ_B^2. The value s is instead an integer scale factor (e.g. 3 or 4), which is the factor by which we want the LR input image I_L to be magnified; i.e. if I_L is of size $N \times M$, the final super-resolved image $\hat{I}_H$ will have a size of $sN \times sM$.

For complexity reason, the SR algorithm later described is applied only on the luminance component Y of the input image I_L (a colorspace transformation from the RGB to the YIQ model is then possibly performed at the

beginning), whereas the color components I and Q are simply upsized by Bicubic interpolation to the final desired size $sN \times sM$. In fact, since humans are more sensitive to changes in the brightness of the image rather than changes in color, it is a common belief that the SR procedure is worthy to be performed only on Y, so reducing the complexity of the algorithm by one third. Hereafter, then, all the image matrices and patch vectors must be intended as collections of pixel luminance values.

As a starting point for our internal dictionary learning procedure, we take the single pyramid depicted in Fig. 2. Here, the top-level is represented by the LR input image itself ($I_0 = I_L$). From it, a finite number of "sub-levels" is created, according the the following relation:

$$I_{-n} = (I_L * B_n) \downarrow_{p^n} , \tag{9}$$

where p, the pyramid cross-level scale factor, is typically a "small" number (e.g. $p = 1.25$). The sub-level image I_{-n} is then a particular rescaled version of the original image I_L (the total rescale factor amount to p^n). As for the variance of the Gaussian kernel B_n to which it is subjected, it can be computed according to the following formula, which is explained in [24]:

$$\sigma^2_{B_n} = n \cdot \sigma^2_B \cdot \log(p) / \log(s) . \tag{10}$$

Once the single pyramid is created, we propose now to interpolate each sub-level I_{-n} by the factor p. The so obtained interpolated level $\mathcal{U}(I_{-n})$, where $\mathcal{U}$ is an upscaling operator s.t. $\mathcal{U}(I) = (I) \uparrow_p$, is an image with the same size as the original non-interpolated level located just above in the pyramid I_{-n+1} (except for possible 1-pixel differences, due to the non-integer interpolation factors). We can then consider the pair constituted by $\mathcal{U}(I_{-n})$ and I_{-n+1} a pair of, respectively, LR and HR training images, from which derive a set of self-examples. By using all the pairs $\{\mathcal{U}(I_{-n}), I_{-n+1}\}$ for $n = 1, \dots NL$, where NL is the chosen number of sub-levels, and sampling at corresponding locations pairs of, respectively, LR and HR patches of equal size $\sqrt{D} \times \sqrt{D}$, we can then form our LR and HR internal dictionary sets: $\mathcal{Y}^l = \{\mathbf{y}^l_i \in \mathcal{R}^D\}_{i=1}^{N_y}$ and $\mathcal{Y}^h = \{\mathbf{y}^h_i \in \mathcal{R}^D\}_{i=1}^{N_y}$. Fig. 5 reports the scheme described, with the "double pyramid" formed by the traditional image cascade and, next to it, a side pyramid of interpolated levels.

B. Gradual upscalings

Once the LR and HR dictionary sets, $\mathcal{Y}^l$ and $\mathcal{Y}^h$, are formed, by populating them with the correspondences of self-examples found in the double pyramid described in Section III-A, the proper SR reconstruction algorithm starts.

The algorithm consists in a multi-pass procedure, where the input image is gradually magnified by an upscale factor equal to the cross-level scale factor p. Given s as the total scale factor to be achieved, the number of necessary passes it then:

$$NP = \lceil \log_p s \rceil . \tag{11}$$

If s is not a power of p, the image super-resolved after NP passes will be over-sized w.r.t. to the targeted dimension ($sM \times sN$); which means that an extra resizing operation is needed. I_0 will be super-resolved into I_1, I_1 into I_2,

and so until obtaining I_{NP}, which will be possibly resized to obtain the SR estimated image $\hat{I}_H$ with the desired dimension. The multi-pass SR procedure is illustrated graphically in Fig. 6.

When generally upscaling the image I_n, this is first interpolated into the image $\mathcal{U}(I_N)$ from which a set of overlapping patches $\mathcal{X}^l = \{\mathbf{x}_i^l\}_{i=1}^{N_x}$ is formed, by scanning it with a sliding window of dimension $\sqrt{D} \times \sqrt{D}$. Each input patch $\mathbf{x}_i^l$ is processed singularly and, after the learning based patch reconstruction procedure, a corresponding HR output patch $\mathbf{x}_i^h$ is produced. By iterating for all patches, we have at the end a set of HR reconstructed patches $\mathcal{X}^h = \{\mathbf{x}_i^h\}_{i=1}^{N_x}$, which will be finally re-assembled to form the upper-level image I_{n+1}. In the following paragraph we describe the patch reconstruction method adopted.

1) Direct mapping of the self-examples via multi-linear regression: While most of the "pyramid-based" SR algorithms in the literature [23]–[25] make use of a neighbor embedding (NE) based procedure to express each input and output patch in terms of combinations of self-examples, we follow instead the direct mapping (DM) approach. As explained in Section II-C, DM aims at learning, thanks to the LR and HR local training sets, a mapping to directly derive the single HR output patch as a function of the related LR input patch.

DM has been employed in SR example-based algorithms using external dictionaries [19], [20], by using in particular the Kernel Ridge Regression (KRR) solution. In this case, the function space $\mathcal{H}$ is what is called a reproducing kernel Hilbert space (RKHS), and the single regression function f_i is seen as an expansion of kernel functions, where Gaussian kernels are typically chosen.

For the sake of simplicity, we believe instead that, since in the case of internal learning the dictionary patches are more pertinent training examples, a simple linear mapping "can do the job". With this goal, $\mathcal{H}$ is taken as a linear function space

$$\mathcal{H} = \left\{ f(\mathbf{x}) = M\mathbf{x} \mid M \in \mathcal{R}^{D \times D}, \mathbf{x} \in \mathcal{R}^D \right\} \tag{12}$$

and the regularized empirical error (7) can be re-expressed as follows:

$$\begin{aligned} M_i &= \underset{M \in \mathcal{R}^{D \times D}}{\arg\min} \sum_{j=1}^{K} \|\mathbf{y}_j^h - M\mathbf{y}_j^l\|_2^2 + \lambda \|M\|_F^2 \\ &= \underset{M \in \mathcal{R}^{D \times D}}{\arg\min} \|Y_i^h - MY_i^l\|_2^2 + \lambda \|M\|_F^2 \ , \end{aligned} \tag{13}$$

where Y_i^l and Y_i^h are the usual, respectively LR and HR, local training sets related to the patch $\mathbf{x}_i^l$. In other words, we are looking for a linear transformation (i.e. the matrix M_i) to be directly applied to the LR input patch. This linear transformation is learned by observing the relations between the LR and HR dictionary patches which are neighbors with $\mathbf{x}_i^l$, according to the machine learning pattern and a regression model, where Y_i^l is the matrix of the *predictor variables* or regressors, and Y_i^h is the matrix of the *response variables*.

As the response variables are vectors and not scalars, we properly speak about *multi-variate regression* (MLR) [32]. The solution to (13) is known, and can be written in a closed-form formula:

$$M_i = Y_i^h {Y_i^l}^T \left(Y_i^l {Y_i^l}^T + \lambda I \right)^{-1} \tag{14}$$

where I is the identity matrix. The equation (14) corresponds exactly to what we called "model generation" (Step 2 of the local learning procedure described in Section II-C). The prediction of the HR unknown patch (Step 3) is the straight application of the linear mapping learned:

$$\mathbf{x}_i^h = M_i \mathbf{x}_i^l \,. \tag{15}$$

Fig. 5 gives a rough depiction of how the DM reconstruction method works in the double pyramid.

2) Patch aggregation and IBP: By learning for each input patch $\mathbf{x}_i^l \in \mathcal{X}^l$ a linear function M_i with the equation (14), and by applying this function to generate the related output patch, we end up with a collection of HR patches $\mathcal{X}^h = \{\mathbf{x}_i^h\}_{i=1}^{N_x}$ that need to be assembled to generate the current upscaling. Before the patch aggregation, the set of reconstructed HR patches $\mathcal{X}^h$ and the equivalent set of LR patches from which they have been originated, $\mathcal{X}^l$, are added to the dictionary as new correspondences of patches to be used in future upscalings: the patches of the two sets, in fact, are equal in number, and a LR-HR relation stands. The update of the dictionary is performed by simple set union, i.e. $\mathcal{Y}^l = \mathcal{Y}^l \cup \mathcal{X}^l$ and $\mathcal{Y}^h = \mathcal{Y}^h \cup \mathcal{X}^h$.

The patches of the input image, and consequently also in the equally-sized output image, are taken with a certain overlap; that means that, when they are re-placed in the original positions, at each pixel location we have a set of different candidate values. We can see the image at this stage as a 3-D image, where the multiple values per pixel are the results of different local observations of the input image (i.e. different patches taken), and therefore contain different partial information about the unknown HR image. We obtain a single output image by convexly combining these candidates at each pixel location: the convex combination is done by simply taking uniform weights, i.e. each candidate is weighted by $1/N_p$, where N_p is the number of overlapping patches contributing to that particular position.

After the image of the new level is formed, by overlapping and averaging, before it is used as a starting image for the next upscaling, it is further refined in an iterative fashion by the *iterative back-projection* (IBP) procedure. IBP is an additional operation adopted by several SR algorithms (e.g. [14], [23]), for which the output super-resolved image, once reconstructed, is "back-projected" to the LR dimension in order to assure it to be consistent with the LR input image, i.e. to assure it to be a plausible estimation of the underlying HR image, and conveniently corrected if errors are observed.

At the iteration t of this refining procedure, the generic reconstructed n-th level I_n^t is first back-projected into an estimated LR image $\hat{I}_L^t$:

$$\hat{I}_L^t = \left(I_n^t * B_n\right) \downarrow_{p^n} \tag{16}$$

where B_n is a Gaussian blur with variance as expressed in (10). The deviation between this LR image found by back-projection and the original LR image is then used to further correct the HR estimated image:

$$I_n^{t+1} = I_n^t + \left(\left(I_L - \hat{I}_L^t\right) \uparrow_{p^n}\right) * b \,, \tag{17}$$

where b is a back-projection filter that locally spreads the differential error.

In Listing 1, our proposed SR procedure is described in a simplified manner, by reporting the pseudocode for the two main routines of the algorithm: "InternalLearning", where the double pyramid is constructed and the dictionary sets of LR and HR patches are initially formed, and "SingleUpscale", that reports the procedure to upscale a generic level I_n to the upper level. To be noted, in particular, on line 15 the *for* loop, which consists of 3 instructions and implements the 3 steps of the local learning based patch reconstruction procedure described in Section II-C: nearest neighbor search, model generation, and prediction.

IV. Experimental Results

In this section we conduct some experiments on the proposed single-image SR algorithm. In particular, in Section IV-A we evaluate the different contributions, in terms of implementation choices, that led to its final formulation summarized in Listing 1. Section IV-B provides some considerations about the complexity, intended both as time and space complexity, of the proposed algorithm. In particular, we evaluate the amount of extra complexity possibly brought by the double pyramid. In Section IV-C, finally, we compare our algorithm with other state-of-the-art methods, by both showing visual comparisons on super-resolved images and reporting quantitative results, according to the *PSNR* and *SSIM* metrics. *PSNR* and *SSIM* values are obtained as measures of the distance between the HR original image, from which the LR input image, for test purposes, has been originated, and the super-resolved image. The image generation model adopted is the one expressed in Equation (1), with the variance of the Gaussian blur σ_B^2 set to 1 in all experiments. The interpolation method used is Bicubic interpolation.

As for the various parameters of the algorithms, they have been "tuned" by empirically looking for their optimal values. Notably, the cross-level scale factor p is taken as $p = 1.25$; as for the patch size, instead, 5×5 patches are sampled from the internal images with a 4-pixel overlap. For each input patch, then, $K = 12$ are selected in the dictionary via NNS.

A. Evaluation of the different contributions

In this section we want to assess the different contributions of our algorithm, which have been discussed in Section III. With respect to the well-known single-image SR algorithm in [23], two main contributions have been presented:

1) The employment of a DM reconstruction method (i.e. MLR), instead of NE;
2) The introduction of a different training and upscaling scheme, i.e. the "double" pyramid.

To singularly evaluate the two "ingredients" above, we test three different procedures (summarized in Table I), which all fall in the category of example-based single-image SR algorithms employing an internal dictionary.

We first start with an algorithm, where a single pyramid is constructed and NE with non-local means (NLM) weights (2) is used as patch reconstruction method ("Algorithm 1"). This algorithm is very close in the spirit to the method in [23], and thus its implementation can be considered as a reference for the mentioned method, except for

Listing 1 Single-image SR via linear mapping of interpolated self-examples

procedure INTERNALLEARNING(I_L, NL, s, p, σ_B^2)
  for $n \leftarrow 1, NL$ **do** ▷ **Create the pyramid levels**
    $\sigma_{B_n}^2 \leftarrow n \cdot \sigma_B^2 \cdot \log(p)/\log(s)$
    $I_{-n} \leftarrow (I_L * B_n) \downarrow_{p^n}$
    $\mathcal{U}(I_{-n}) \leftarrow (I_{-n}) \uparrow_p$
  end for

  for $n \leftarrow 1, NL$ **do** ▷ **Populate the internal dict.**
    Sample patches from $\mathcal{U}(I_{-n})$ and add to $\mathcal{Y}^l$
    Sample patches from I_{-n+1} and add to $\mathcal{Y}^h$
  end for
end procedure

procedure SINGLEUPSCALE(I_n, p, $\mathcal{Y}^l$, $\mathcal{Y}^h$)
  $\mathcal{U}(I_n) \leftarrow (I_n) \uparrow_p$ ▷ **Upscale the current level**
  Extract LR patches from $\mathcal{U}(I_n) \rightarrow \mathcal{X}^l = \{\mathbf{x}_i^l\}_{i=1}^{N_x}$

  for $i \leftarrow 1, N_x$ **do** ▷ **Single patch reconstructions**
    Find the local train. sets of $\mathbf{x}_i^l$ by NNS $\rightarrow Y_i^l, Y_i^h$
    $M_i \leftarrow Y_i^h {Y_i^l}^T \left(Y_i^l {Y_i^l}^T + \lambda I \right)^{-1}$
    $\mathbf{x}_i^h \leftarrow M_i \mathbf{x}_i^l$
  end for

  Form I_{n+1} with the constr. patches $\mathcal{X}^h = \{\mathbf{x}_i^h\}_{i=1}^{N_x}$
  Refine I_{n+1} by IBP
  $\mathcal{Y}^l \leftarrow \mathcal{Y}^l \cup \mathcal{X}^l$ ▷ **Update the LR dictionary**
  $\mathcal{Y}^h \leftarrow \mathcal{Y}^h \cup \mathcal{X}^h$ ▷ **Update the HR dictionary**
end procedure

TABLE I
SUMMARY OF THE INTERNAL-DICTIONARY PROCEDURES CONSIDERED TO EVALUATE THE DIFFERENT CONTRIBUTIONS OF OUR ALGORITHM.

	Rec. Method	**Double pyr.**
Algorithm 1	NE	No
Algorithm 2	DM	No
Proposed	DM	Yes

possible slight differences in the code configuration and the choice of the parameters. With respect to Algorithm 1, "Algorithm 2" uses a DM patch reconstruction method instead of NE, i.e. the MLR method described in Section III-B that linearly maps each LR patch into the related HR patch. The finally proposed algorithm introduces then the double pyramid scheme, featuring the side pyramid of interpolated levels. Table II presents the *PSNR* and *SSIM* values for all the considered algorithms, when tested on seven input images and for two different scale factors ($s = 3, 4$).

As we can observe from the table, from Algorithm 1 to the Proposed algorithm we have almost always a progressive consistent improvement in the SR performance, with our finally proposed procedure presenting an average gain of about 0.22 dB w.r.t. to the traditional scheme based on a single pyramid and NE (Algorithm 1). This gain can be appreciated when observing the output images. Fig. 7 shows in fact the visual results obtained with the three different procedures, for two super-resolved images.

As we can observe from the zoomed-in areas of the images, with the finally proposed algorithm, we are able to produce finer details (see the branch of the tree behind the bird, or the texture of the hat of the woman). In particular, by observing the woman's hat, we can appreciate a sort of progress in the outcome of the three algorithms: Algorithm 1 gives a pretty blurred result; Algorithm 2, thanks to the use of DM in the place of NE, shows instead more regular structures; the edges and the shapes of these structures look even sharper with the Proposed algorithm, where the DM functions have been computed on the interpolated patches of the double pyramid.

B. Considerations on the algorithm complexity

In order to evaluate the time complexity of the algorithm, we ran two versions of it, featuring the usual single pyramid and our double pyramid, for two different scale factors (3 and 4). The single-pyramid implementation gives an idea of the complexity of several algorithms in the literature (e.g. [23]–[25]). The goal of this test is to evaluate the penalty brought by the use of our double pyramid, in terms of increased complexity. Table III summarizes the results of the tests made.

As we can see from Table III, the double pyramid leads to an extra complexity in the order of about 10%. The increase in the computational cost is mainly due to the fact that bigger LR vectors are processed (LR patches have the same size as HR patches). Nevertheless, we believe that the complexity increase is still within a tolerable extent.

TABLE II

PERFORMANCE RESULTS (*PSNR* AND *SSIM* VALUES) OF THE THREE DIFFERENT ALGORITHMS USING AN INTERNAL DICTIONARY, WHEN SUPER-RESOLVING SEVERAL IMAGES FOR SCALE FACTORS OF 3 AND 4.

		Algorithm 1		**Algorithm 2**		**Proposed**	
Image	**Scale**	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*
Bike	3	23.13	0.792	23.20	0.796	23.30	0.799
Bird	3	33.71	0.884	34.06	0.890	34.13	0.890
Butterfly	3	27.08	0.833	27.38	0.840	27.56	0.841
Hat	3	29.95	0.651	30.11	0.655	30.11	0.655
Head	3	32.43	0.633	32.47	0.667	32.43	0.668
Lena	3	30.68	0.770	30.91	0.775	30.89	0.775
Woman	3	30.33	0.862	30.41	0.866	30.52	0.867
Avg gain w.r.t. A1		-	-	0.18	0.009	0.23	0.010
Bike	4	21.58	0.685	21.53	0.688	21.63	0.689
Bird	4	30.61	0.810	30.89	0.824	31.01	0.826
Butterfly	4	24.68	0.763	24.71	0.772	25.05	0.775
Hat	4	28.54	0.553	28.60	0.556	28.53	0.555
Head	4	31.17	0.556	31.28	0.591	31.23	0.590
Lena	4	29.02	0.681	29.22	0.687	29.28	0.689
Woman	4	27.60	0.778	27.90	0.793	27.96	0.793
Avg gain w.r.t. A1		-	-	0.13	0.012	0.21	0.013

TABLE III

AVERAGE RUNNING TIME OF THE ALGORITHM, FOR TWO SCALE FACTORS, WITH SINGLE OR DOUBLE PYRAMID. TIME IS EXPRESSED IN SECONDS.

Scale	**LR size**	**N. Passes**	**Single Pyr.**	**Double Pyr.**	ΔT
3	85×85	5	75	83	+10.7%
4	64×64	7	111	120	+8.1%

To evaluate the space complexity, i.e. memory storage requirement, we can instead compute an estimation of the size of the internal dictionary. Here, the only discriminating factor between the single and the double pyramid is the size of the LR patches. In fact, in the case of double pyramid, both patches have the same size, whereas the LR patches are p^2 smaller in the case of single pyramid (where p is the small scale factor used at each pass). Given this, it is easy to compute the ratio between the two dictionary sizes:

$$R = \frac{2}{1 + \frac{1}{s^2}} \ . \tag{18}$$

In our tests p was set to 1.25, which corresponds to a 22% increase of the storage requirement.

C. Comparison with state-of-the-art algorithms

In this section we perform a comparative assessment of our method with other single-image SR algorithms in the state-of-the art, by extending the comparison also to SR methods based on external dictionaries. For this purpose, we consider Bicubic interpolation as a reference for traditional analytical interpolation methods, and four other example-based SR algorithms. The characteristic of the latter, as well as those ones of our proposed method, are summarized in Table IV.

TABLE IV
SUMMARY OF OUR METHOD AND THE FOUR EXAMPLE-BASED SR ALGORITHMS CONSIDERED FOR THE COMPARISON.

Method	**Reference**	**Dictionary**	**Rec. Method**
LLE-based NE	Chang et al. [9]	External	NE with LLE weights (3)
Nonnegative NE	Bevilacqua et al. [33]	External	NE with NN weights (4)
Sparse SR	Yang et al. [14]	External	Sparse coding
Pyramid	Glasner et al. [23]	Internal, single pyramid	NE with NLM weights (2)
Proposed	**This paper**	**Internal, double pyramid**	**DM via Multi-linear Regress.**

For the first three methods in the table, the original code of the respective authors, possibly slightly modified to make the comparison fair, has been used. For the "pyramid" method of [23], instead, the third-party code of the authors of [24] has been adopted. Table V reports the performance results of the algorithms considered (*PSNR* and *SSIM* values), with the usual set of seven test images, and 3 and 4 as scale factors.

Table V shows clearly that our method outperforms the other algorithms in terms of objective quality of the super-resolved images, for all the images and the two scale factors considered. The results are confirmed when observing the visual comparisons (Fig. 8, Fig. 9, and Fig. 10).

From the visual results presented, we can see that methods based on external dictionaries ("LLE-based NE", "NN NE", and "Sparse SR") often present blurring and ringing artifacts (see e.g. the *Bike* images in Fig. 9). Results for the two algorithms based on an internal dictionary (the "pyramid" algorithm of [23] and ours), instead, are certainly more pleasant at sight, presenting sharp edges and smooth artifact-free results in the regions with no texture. However, the algorithm of [23] in some cases shows results that look somehow "artificial", with over-smoothed areas or unnatural edges (see Lena's eye in Fig. 10e, whose shape is deformed). Our algorithm succeeds also in avoiding these undesired effects.

TABLE V

PERFORMANCE COMPARISON (*PSNR* AND *SSIM* VALUES) WITH OTHER STATE-OF-THE-ART METHODS, WHEN SUPER-RESOLVING SEVERAL IMAGES FOR SCALE FACTORS OF 3 AND 4.

		Bicubic		**LLE-based NE**		**NN NE**		**Sparse SR**		**Pyramid**		**Proposed**	
Image	**Scale**	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*
Bike	3	20.84	0.640	22.38	0.754	23.02	0.784	22.71	0.771	22.81	0.772	23.30	0.799
Bird	3	29.88	0.820	32.11	0.834	33.37	0.864	32.90	0.858	32.18	0.862	34.13	0.890
Butterfly	3	21.75	0.712	24.27	0.774	26.36	0.823	24.92	0.787	26.63	0.825	27.56	0.841
Hat	3	27.27	0.536	29.24	0.619	29.65	0.632	29.21	0.602	29.66	0.626	30.11	0.655
Head	3	31.02	0.583	32.05	0.626	32.28	0.638	32.23	0.653	31.69	0.629	32.43	0.668
Lena	3	28.03	0.670	29.78	0.729	30.52	0.754	30.06	0.738	30.28	0.748	30.89	0.775
Woman	3	26.17	0.778	28.27	0.806	29.35	0.840	29.02	0.815	29.89	0.857	30.52	0.867
Average		26.42	0.677	28.30	0.734	29.22	0.762	28.72	0.746	29.02	0.760	29.85	0.785
Bike	4	19.83	0.536	20.75	0.634	21.26	0.667	20.93	0.651	21.46	0.681	21.63	0.689
Bird	4	28.09	0.738	29.25	0.741	30.41	0.786	30.00	0.787	30.10	0.803	31.01	0.826
Butterfly	4	20.30	0.637	21.83	0.682	23.41	0.738	22.18	0.701	24.44	0.761	25.05	0.775
Hat	4	26.30	0.462	27.48	0.505	27.85	0.530	27.31	0.506	28.41	0.544	28.53	0.555
Head	4	30.07	0.516	30.71	0.537	30.92	0.556	30.85	0.573	30.86	0.575	31.23	0.590
Lena	4	26.85	0.593	27.84	0.628	28.52	0.670	27.98	0.649	28.70	0.668	29.28	0.689
Woman	4	24.61	0.697	25.84	0.696	26.76	0.752	26.18	0.732	27.20	0.777	27.96	0.793
Average		25.15	0.597	26.24	0.632	27.02	0.671	26.49	0.657	27.31	0.687	27.81	0.702

V. CONCLUSION

In this paper we presented a novel single-image single-image (SR) method, which belongs to the family of example-based SR algorithms, using a dictionary of patches in the upscaling procedure. The algorithm originally makes use of a "double pyramid" of images, built starting from the input image itself, to extract the dictionary patches (thus called "self-examples"), and employs a regression-based method to directly map the low-resolution (LR) input patches into their related high-resolution (HR) output patches. When compared to other state-of-the-art algorithms, our proposed algorithm shows the best performance, both in terms of objective metrics and subjective visual results. As for the former, it presents considerable gains in *PSNR* and *SSIM* values. When observing the super-resolved images, also, it turns out to be the most capable in producing fine artifact-free HR details. The algorithm does not rely on extra information, since making use of an internal dictionary automatically "self-adapted" to the input image content, and also the few parameters have proven to be easy to tune. This makes it a particularly attractive method for SR upscaling purposes.

REFERENCES

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-Resolution Image Reconstruction: A Technical Overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, 5 2003.

[2] T. S. Huang and R. Y. Tsai, "Multiframe image restoration and registration," *Adv. Comput. Vis. Image Process.*, vol. 1, no. 7, pp. 317–339, 1984.

[3] X. Li and M. T. Orchard, "New Edge-Directed Interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[4] M. F. Tappen, B. C. Russell, and W. T. Freeman, "Exploiting the Sparse Derivative Prior for Super-Resolution and Image Demosaicing," in *Proc. IEEE Int. Work. Statist. Comput. Theories Vis.*, 2003.

[5] R. Fattal, "Image Upsampling via Imposed Edge Statistics," *ACM Trans. Graphics*, vol. 26, no. 3, pp. 1–8, Jul. 2007.

[6] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, 2011, pp. 449–456.

[7] K. Zhang, X. Gao, D. Tao, and X. Li, "Single Image Super-Resolution With Non-Local Means and Steering Kernel Regression," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4544–4556, 2012.

[8] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-Based Super-Resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, 2002.

[9] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-Resolution Through Neighbor Embedding," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, vol. 1, 2004, pp. 275–282.

[10] W. Fan and D.-Y. Yeung, "Image Hallucination Using Neighbor Embedding over Visual Primitive Manifolds," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, 7 2007, pp. 1–7.

[11] T.-M. Chan, J. Zhang, J. Pu, and H. Huang, "Neighbor embedding based super-resolution algorithm through edge detection and feature selection," *Pattern Recogn. Lett.*, vol. 30, no. 5, pp. 494–502, 4 2009.

[12] B. Li, H. Chang, S. Shan, and X. Chen, "Locality preserving constraints for super-resolution with neighbor embedding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 11 2009, pp. 1189–1192.

[13] X. Gao, K. Zhang, D. Tao, and X. Li, "Joint learning for single-image super-resolution via a coupled constraint," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 469–480, Feb. 2012.

[14] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image Super-Resolution Via Sparse Representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861 –2873, 11 2010.

[15] J. Wang, S. Zhu, and Y. Gong, "Resolution enhancement based on learning the sparse association of image patches," *Pattern Recogn. Lett.*, vol. 31, pp. 1–10, 1 2010.

[16] X. Gao, K. Zhang, D. Tao, and X. Li, "Image Super-Resolution With Sparse Neighbor Embedding," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3194–3205, 2012.

[17] K. Zhang, X. Gao, D. Tao, and X. Li, "Multi-scale dictionary for single image super-resolution," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, Jun. 2012, pp. 1114–1121.

[18] J. Yang, Z. Lin, and S. Cohen, "Fast Image Super-Resolution Based on In-Place Example Regression," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, Jun. 2013, pp. 1059–1066.

[19] Y. Tang, P. Yan, Y. Yuan, and X. Li, "Single-image super-resolution via local learning," *Int.J. Mach. Learning Cybern.*, vol. 2, pp. 15–23, 2011.

[20] K. I. Kim and Y. Kwon, "Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, 2010.

[21] G. Freedman and R. Fattal, "Image and Video Upscaling from Local Self-Examples," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 1–10, 2010.

[22] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with multiscale similarity learning," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, no. 10, pp. 1648–1659, Oct. 2013.

[23] D. Glasner, S. Bagon, and M. Irani, "Super-Resolution from a Single Image," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, 10 2009, pp. 349–356.

[24] C.-Y. Yang, J.-B. Huang, and M.-H. Yang, "Exploiting Self-Similarities for Single Frame Super-Resolution," in *Proc. Asian Conf. Comp. Vis. (ACCV)*, Nov. 2010.

[25] M.-C. Yang, C.-H. Wang, T.-Y. Hu, and Y.-C. F. Wang, "Learning context-aware sparse representation for single image super-resolution," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2011, pp. 1349–1352.

[26] M. Zontak and M. Irani, "Internal Statistics of a Single Natural Image," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, 2011, pp. 977–984.

[27] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *Proc. British Mach. Vis. Conf. (BMVC)*. Guildford, UK: BMVA Press, Sep. 2012, pp. 135.1–135.10.

[28] S. Baker and T. Kanade, "Limits on Super-Resolution and How to Break Them," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1167–1183, 2002.

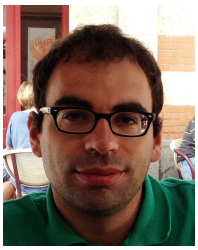
[29] A. Buades, B. Coll, and J.-M. Morel, "A Non-Local Algorithm for Image Denoising," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. (CVPR)*, vol. 2, Washington, DC, USA, 2005, pp. 60–65.

[30] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 12 2000.

[31] R. Timofte, V. D. Smet, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, Dec. 2013.

[32] H. Hyötyniemi, "Multivariate regression - Techniques and tools," Helsinki University of Technology, Control Engineering Laboratory, Helsinki, Finland, Tech. Rep. Report 125, 2001.

[33] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, "Super-resolution using Neighbor Embedding of Back-projection residuals," in *Proc. Int. Conf. Digit. Signal Process. (DSP)*, Santorini, Greece, Jul. 2013.

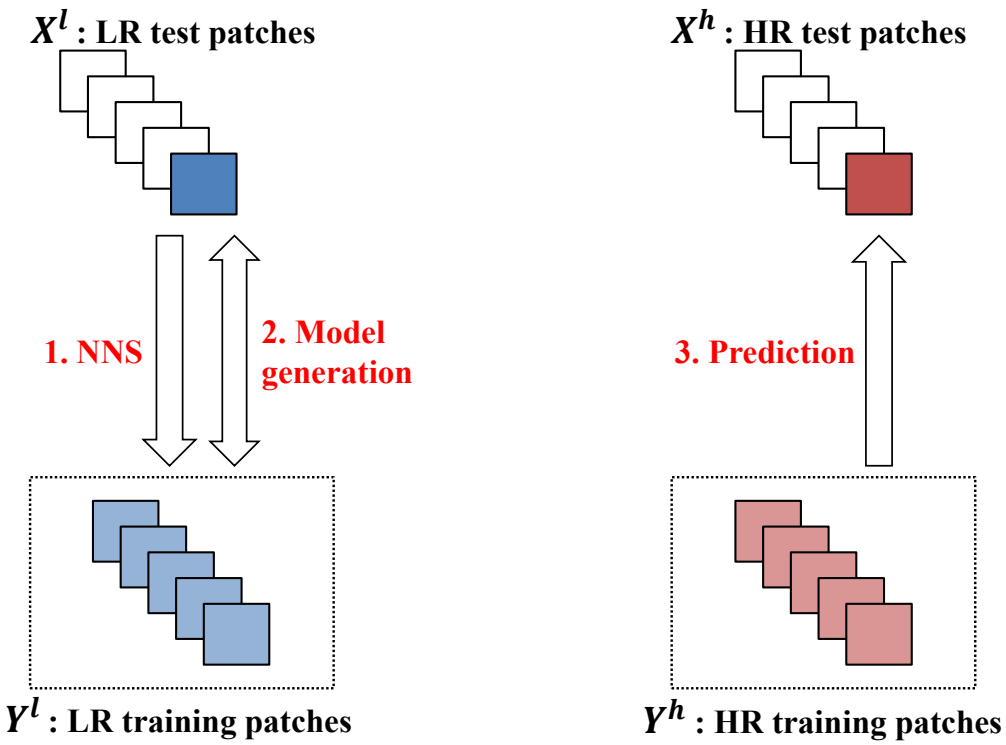


Marco Bevilacqua Marco Bevilacqua is a Postdoctoral Research Fellow at IMT Institute for Advanced Studies Lucca, Italy. He received the Laurea (B.S. equivalent degree) and Laurea Specialistica (M.S. equivalent degree) from University of Florence, Italy, in 2006 and 2009 respectively. From February 2010 to January 2011, he was a research fellow at the Italian National Research Council (C.N.R.). From February 2011 to March 2014, as a Ph.D. student of University of Rennes 1 (France), he pursued his Ph.D. studies in Computer Science, by carrying out a thesis project co-funded by INRIA (Institut National de Recherche en Informatique et en Automatique) and Alcatel-Lucent Bell Labs France. He finally received his Ph.D. degree in June 2014. His research interests lie in signal processing, particularly for images, computer vision, and machine learning techniques.

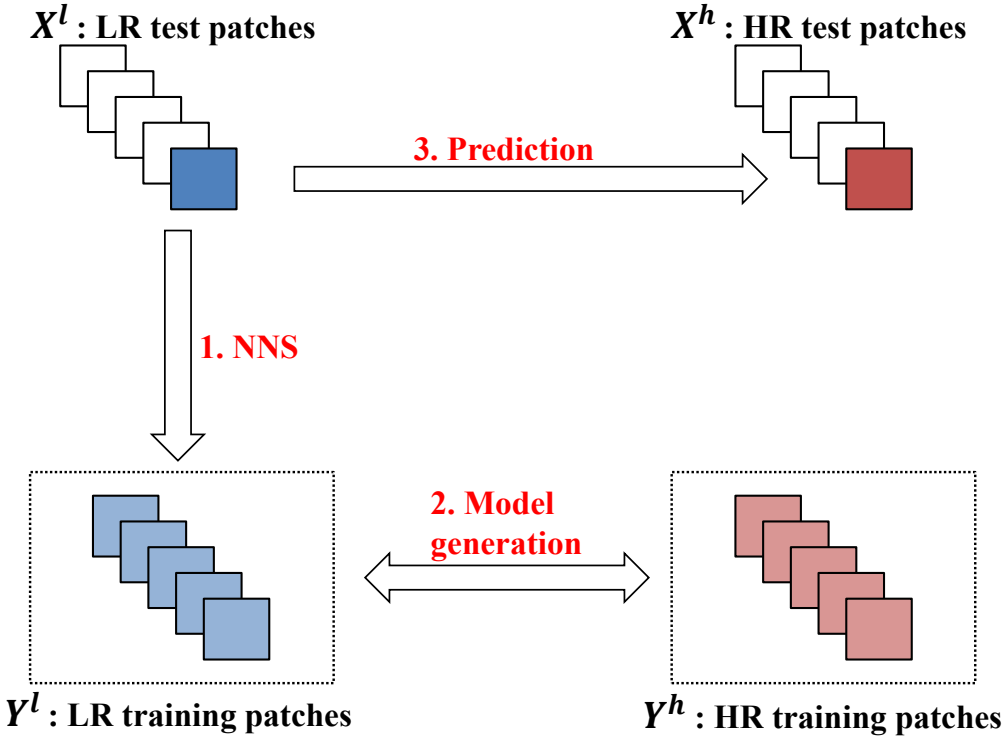
Aline Roumy Aline Roumy received the Engineering degree from Ecole Nationale Suprieure de l'Electronique et de ses Applications (ENSEA), Cergy, France in 1996, the Master degree in June 1997 and the Ph.D. degree in September 2000 from the University of Cergy-Pontoise, France. During 2000-2001, she was a research associate at Princeton University, Princeton, NJ. On November 2001, she joined INRIA, Rennes, France as a research scientist. She has held visiting positions at Eurecom and Berkeley University. Her current research and study interests include the area of statistical signal and image processing, coding theory and information theory. She has been a Technical Program Committee member and session chair at several international conferences, including ISIT, ICASSP, Eusipco, WiOpt, Crowncom. She is currently serving as a member of the French National University Council (CNU 61). She received the 2011 "Francesco Carassa" best paper award.

Christine Guillemot Christine Guillemot, IEEE fellow, is "Director of Research" at INRIA, head of a research team dealing with image and video modeling, processing, coding and communication. She holds a Ph.D. degree from ENST (Ecole Nationale Superieure des Telecommunications) Paris, and an "Habilitation for Research Direction" from the University of Rennes. From 1985 to Oct. 1997, she has been with FRANCE TELECOM, where she has been involved in various projects in the area of image and video coding for TV, HDTV and multimedia. From Jan. 1990 to mid 1991, she has worked at Bellcore, NJ, USA, as a visiting scientist. She has (co)-authored 15 patents, 8 book chapters, 50 journal papers and 140 conference papers. She has served as associated editor (AE) for the IEEE Trans. on Image processing (2000-2003), for IEEE Trans. on Circuits and Systems for Video Technology (2004-2006) and for IEEE Trans. On Signal Processing (2007-2009). She is currently AE for the Eurasip journal on image communication and member of the editorial board for the IEEE Journal on selected topics in signal processing (2013-2015). She is a member of the IEEE IVMSP technical committees.

Marie-Line Alberi Morel Dr. Marie-Line Alberi Morel is a researcher since 2001 at Alcatel-Lucent Bell Labs in Nozay, France. She received a M.S. degree in electronics and signal processing in 1995, and a Ph.D. degree in the signal processing field in 2001 from University of Paris Sud Orsay, France. During her career at Alcatel-Lucent, she has contributed to various radio projects on deployment and dimensioning studies for WLAN and small cell networks. She has conducted research on DVB-SH mobile broadcast network ("Télévision Mobile Sans Limite" French project). Later, she worked on the design of optimized solutions of video services delivery over mobile wireless networks and over converging 3GPP EMBMS and DVB NGH broadcasting networks (Adaptable, Robust, Streaming Solutions and Mobile Multi Media French projects). Currently, her research focuses on the video processing field, particularly on super resolution algorithms based on machine learning techniques and on the source coding. She also serves as an associate professor at Marne-La-Vallée University, France.



(a) Neighbor Embedding



(b) Direct Mapping

Fig. 4. Operating schemes of the local learning based reconstruction procedure, for the two approaches described (NE and DM). From the image it is clear how NE exploits the intra-space relationships, by learning a model among the LR patches and applying it on the HR patches. DM, instead, exploits the "horizontal" relationships, by attempting to learn the mapping between the two spaces.

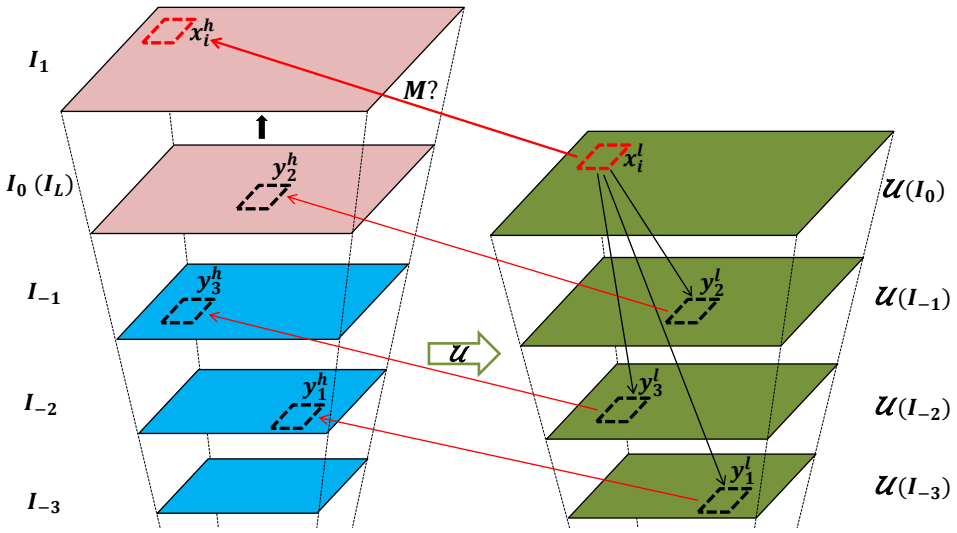


Fig. 5. Creation of the "double pyramid" and search of self-examples throughout it. The figure concerns the upscaling of I_0 to I_1. Given a reference patch in the interpolated version of the starting image $\mathcal{U}(I_0)$, $\mathbf{x_i^l}$, 3 LR neighbors are found in the "side pyramid" of interpolated levels ($\mathbf{y_1^l}, \mathbf{y_2^l}, \mathbf{y_3^l}$). Thanks to these and the corresponding HR neighbors ($\mathbf{y_1^h}, \mathbf{y_2^h}, \mathbf{y_3^h}$), a linear function M is meant to be learned to directly map $\mathbf{x_i^l}$ into its corresponding HR output patch $\mathbf{x_i^h}$.

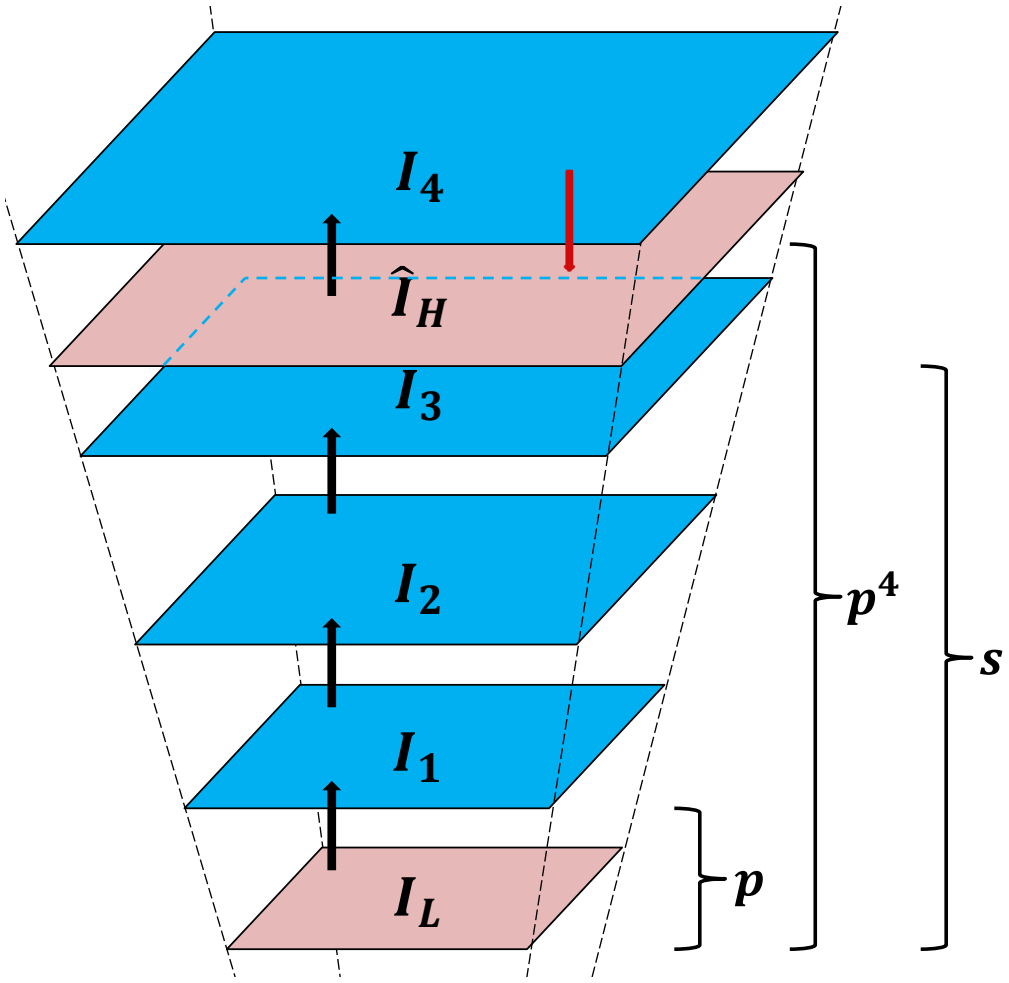
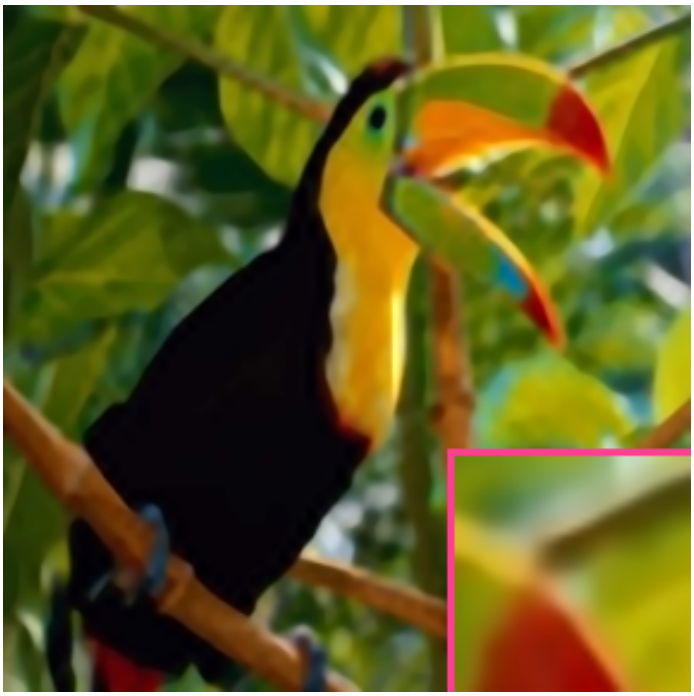
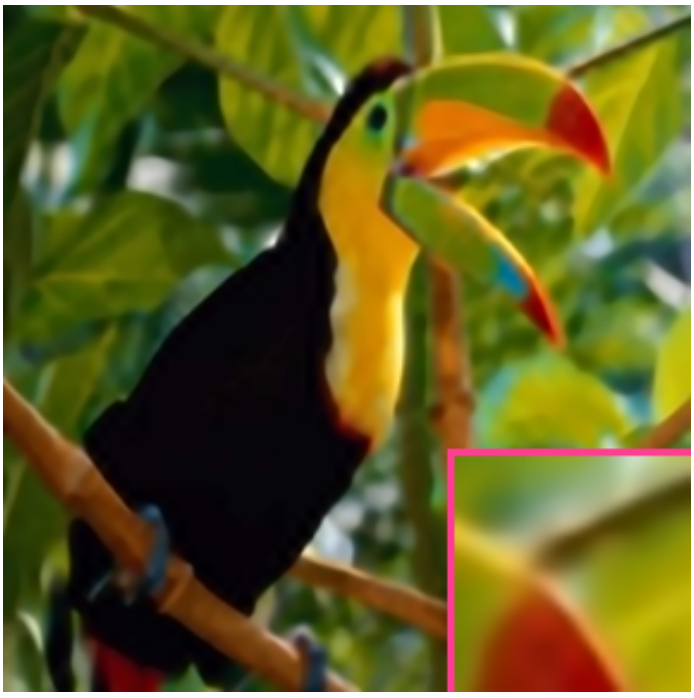


Fig. 6. Estimation of the HR image by gradual upscalings. The original image I_L, by 4 upscalings by a factor of p, is super-resolved into I_4, that exceeds in dimension the desired scale factor s. $\hat{I}_H$ is obtained by a finally resizing operation.



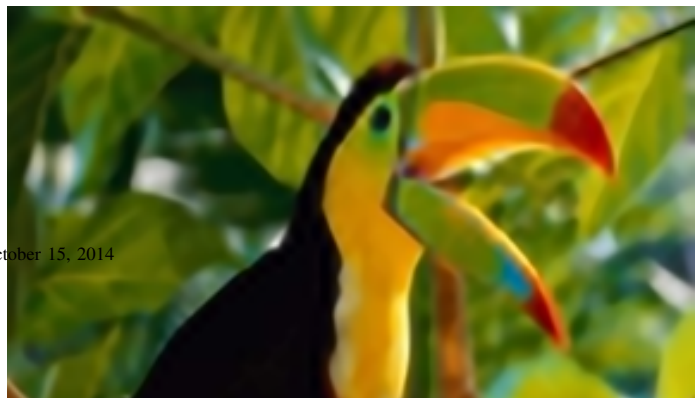
Algorithm 1

Algorithm 1



Algorithm 2

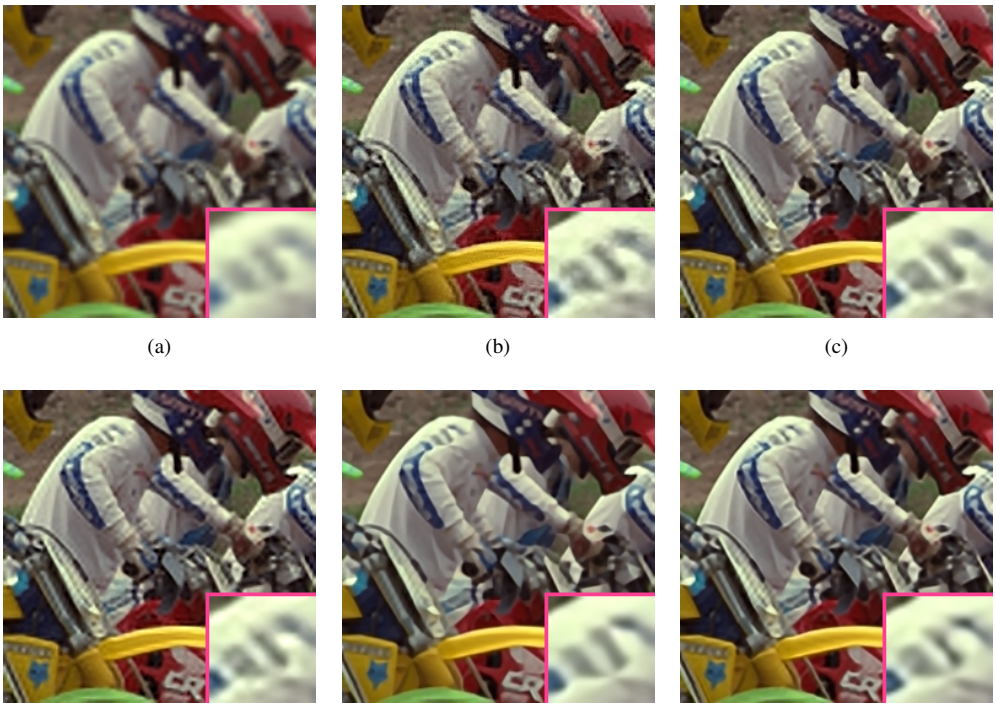
Algorithm 2



(a) (b) (c)

(d) (e) (f)

Fig. 8. Comparative results with zoomed-in areas for *Butterfly* magnified by a factor of 3. The methods considered are: (a) Bicubic interpolation, (b) LLE-based NE, (c) NN NE, (d) Sparse SR, (e) Pyramid, (f) Proposed.



(a) (b) (c)

(d) (e) (f)

Fig. 9. Comparative results with zoomed-in areas for *Bike* magnified by a factor of 3. The methods considered are: (a) Bicubic interpolation, (b) LLE-based NE, (c) NN NE, (d) Sparse SR, (e) Pyramid, (f) Proposed.

(a) (b) (c)

(d) (e) (f)

Fig. 10. Comparative results with zoomed-in areas for *Lena* magnified by a factor of 3. The methods considered are: (a) Bicubic interpolation, (b) LLE-based NE, (c) NN NE, (d) Sparse SR, (e) Pyramid, (f) Proposed.