

NEIGHBOR EMBEDDING BASED SINGLE-IMAGE SUPER-RESOLUTION USING SEMI-NONNEGATIVE MATRIX FACTORIZATION

Marco Bevilacqua* Aline Roumy* Christine Guillemot* Marie-Line Alberi Morel†

* IRISA-INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France

† Alcatel-Lucent, Bell Labs France, Route de Villejust, 91620 Nozay, France

ABSTRACT

This paper describes a novel method for single-image super-resolution (SR) based on a neighbor embedding technique which uses Semi-Nonnegative Matrix Factorization (SNMF). Each low-resolution (LR) input patch is approximated by a linear combination of nearest neighbors taken from a dictionary. This dictionary stores low-resolution and corresponding high-resolution (HR) patches taken from natural images and is thus used to infer the HR details of the super-resolved image. The entire neighbor embedding procedure is carried out in a feature space. Features which are either the gradient values of the pixels or the mean-subtracted luminance values are extracted from the LR input patches, and from the LR and HR patches stored in the dictionary. The algorithm thus searches for the K nearest neighbors of the feature vector of the LR input patch and then computes the weights for approximating the input feature vector. The use of SNMF for computing the weights of the linear approximation is shown to have a more stable behavior than the use of LLE and lead to significantly higher PSNR values for the super-resolved images.

Index Terms— Super-resolution, neighbor embedding, Semi-nonnegative Matrix Factorization

1. INTRODUCTION

Super-resolution (SR) refers to the task of producing a high-resolution (HR) image from one or more low-resolution (LR) images. In this work we focus on the single-image problem, whose aim is to synthesize an “enhanced” enlargement of a target image with possibly better results than traditional interpolation techniques.

Single-image Super-Resolution is often referred to as *Example-based Super-Resolution* [1, 2], as the prior information required in order to estimate the missing HR details is given in the form of *examples*, i.e. learned pairs of LR and HR patches (sub-windows of image) that compose a dictionary. The learned pairs of patches are usually taken from external HR images and degraded (blurred and conveniently down-sized) versions of them. As a first step of any example-based SR algorithm, the target image is divided into patches of the same size of the LR patches in the dictionary; then, each LR

input patch is compared to the stored LR patches and, once the nearest patch among these is found, the corresponding HR patch is finally taken as the output. A variation to this procedure is presented in [3] and in some other SR methods based on sparse representations (e.g. [4, 5]): instead of selecting from the dictionary only one patch, several patches are taken into account and contribute simultaneously to the generation of a single HR output patch. In particular, in [3] the authors propose a single-image SR algorithm, based on the concept of neighbor embedding and originally inspired by a method for data dimensionality reduction called *Locally Linear Embedding* (LLE) [6]. The basic assumption is that a patch in the LR target image and the corresponding HR unknown patch share similar neighborhood structures: as a consequence of that, once the LR patch is expressed as the linear combination of a certain number of its neighbors taken from the dictionary, the output patch can be reconstructed by using the HR patches in the dictionary corresponding to the neighbors selected, and combining them in the same way. This algorithm is shown to be suitable for the SR problem, but its performance is sensitive to the number of neighbors chosen, that appears as a parameter difficult to properly set.

We propose a new neighbor embedding method based on Semi-nonnegative Matrix Factorization (SNMF) [7]. In LLE the weights are constrained to sum up to one, but no constraint is specified for their sign. This might explain the unstable results observed in [3], since possible negative weights can lead to having subtractive combinations of patches, which is counterintuitive. Instead, we suggest to compute the weights with a non-negative constraint, to be compatible with the intuitive notion of “combining parts to form a whole” (from [8]). In neighbor embedding based SR, both the LR and the HR patches are represented by *feature vectors*, given by the concatenation of some, possibly different, features of their pixels, and all the SR steps are performed by using these vectors. We thus discuss the issue of the choice of the features, and propose to use the centered luminance values of the pixels to represent the LR patches, instead of the gradient as in [3].

The paper is organized as follows. Section 2 presents neighbor embedding as a possible approach for super-resolution. Then, Section 3 describes our algorithm, particularly the new method for computing the neighbor combination and the

choice of the features. In sections 4 and 5, experimental results are finally reported and commented.

2. SUPER-RESOLUTION VIA NEIGHBOR EMBEDDING

The Neighbor Embedding (NE) algorithm of Chang et al. [3] is the starting point for this work. As said, its peculiarity is that several patches are taken from the dictionary to represent a single input patch; it follows that multiple patches, i.e. their HR counterparts in the dictionary, are combined together to generate an output HR patch. In the algorithm, each patch is represented by a feature vector: we extract from any single pixel of the patch one or more features; the patch vector is then obtained by the simple concatenation of the features of its pixels. The neighbor search, the weight computation and the HR patch reconstruction steps are all performed with respect to the feature spaces chosen: once the HR feature vectors are reconstructed, we finally “reverse” the features, so obtaining the actual HR output patches. In [3], for instance, a 4-value gradient and the centered luminance pixel values are used as features, respectively for the LR and HR patches. In Table 1 the notations used in the paper for the corresponding feature vectors are reported.

$\mathcal{X}_d = \{\mathbf{x}_d^j\}_{j=1}^{N_d}$	LR patches in the dictionary
$\mathcal{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{N_t}$	patches in the LR target image
$\mathcal{Y}_d = \{\mathbf{y}_d^j\}_{j=1}^{N_d}$	HR patches in the dictionary
$\mathcal{Y}_t = \{\mathbf{y}_t^i\}_{i=1}^{N_t}$	patches in the HR target image
<i>(Note: all patches are intended as patch feature vectors.)</i>	

Table 1: Notation used for different types of patch feature vectors.

The NE-based SR approach proceeds as follows.

1. For each LR patch feature vector $\mathbf{x}_t^i \in \mathcal{X}_t$
 - (a) Find its K nearest neighbors (NN) in \mathcal{X}_d in terms of Euclidean distance:

$$\mathcal{N}_i = \arg \min_{\mathbf{x}_d^j \in \mathcal{X}_d}^K \left\| \mathbf{x}_t^i - \mathbf{x}_d^j \right\|^2 .$$

- (b) Compute according to some criteria the weights of the linear combination that approximates \mathbf{x}_t^i with the selected neighbors, i.e. the K weights $\{w_{ij}\}_{j=1}^K$ such that:

$$\mathbf{x}_t^i \approx \sum_{\mathbf{x}_d^j \in \mathcal{N}_i} w_{ij} \mathbf{x}_d^j .$$

In [3], for instance, the original Neighbor Embedding algorithm takes from LLE the method used to “embed” the neighbors. For each input patch, the weights related to its neighbors are found by solving the following minimization problem:

$$\mathbf{w}^i = \arg \min_{\mathbf{w}} \left\| \mathbf{x}_t^i - X_d^i \mathbf{w} \right\|^2 \quad \text{s.t.} \quad \mathbf{1}^T \mathbf{w} = 1 , \quad (1)$$

where \mathbf{w}^i is the $K \times 1$ weight vector related to the input patch \mathbf{x}_t^i , and X_d^i is a matrix with all its neighbors disposed as columns. (1) can be seen as a Constrained Least Squares (LS) problem, whose solution is provided in [3, 6]. Therefore, in this paper we refer to the weights in [3] as the “LS weights”.

- (c) Apply the same weights for the reconstruction of the output HR patch feature vector \mathbf{y}_t^i with the corresponding neighbors in \mathcal{Y}_d :

$$\mathbf{y}_t^i = \sum_{\mathbf{y}_d^j \in \mathcal{H}(\mathcal{N}_i)} w_{ij} \mathbf{y}_d^j .$$

where $\mathcal{H}(\mathcal{N}_i)$ indicates the corresponding set of the neighborhood \mathcal{N}_i in the HR dictionary \mathcal{Y}_d .

2. Once all the HR patch feature vectors are generated, we reverse the features and combine the obtained pixel-based patches to form the output image. Since the LR patches are taken from the input image with some overlap, also the LR patches will be mutually overlapped: the final image is then obtained by simply averaging the pixel values in the common regions.

The key points of the neighbor embedding scheme are the criterion chosen to compute the weights of each neighbor combination and the features used to represent the patches. Results from the original Neighbor Embedding SR algorithm, which is characterized by the so-called LS weights and gradient-based features, show that the error between the super-resolved image and the ground-truth reaches a minimum value for a certain number of neighbors K (see Figure 8 in [3]). However, this minimum is not always reached for the same K but seems to depend on the target image. This might be due to the constraint used in [3]. In fact, in [3] the weights are constrained to sum to one in order to make the reconstruction independent of a frame of reference. This can lead to have negative weights, i.e. the final output HR patches are obtained from both additive and subtractive combinations of patches. We believe that having a non-negativity constraint on the computed weights can help in having a more regular behavior of the performance of the neighbor embedding algorithm, i.e. an error strictly decreasing with K .

In Section 3.1 we derive from SNMF another method to compute the neighbor embedding, that assures the weights to be non-negative. In Section 3.2 we discuss the issue of the features and the other parameters of the algorithm.

3. OUR SNMF-BASED ALGORITHM

3.1. Non-negative weight computation

As we look for non-negative weights, (1) can be changed as follows:

$$\mathbf{w}^i = \arg \min_{\mathbf{w}} \left\| \mathbf{x}_t^i - X_d^i \mathbf{w} \right\|^2 \quad \text{s.t.} \quad \mathbf{w} \geq 0 . \quad (2)$$

(2) can be efficiently approximated by Semi-nonnegative Matrix Factorization (SNMF) [7]. In fact, SNMF is a method to perform a factorization of a matrix, where, differently from “full” Nonnegative Matrix Factorization (NMF) [8], only one factor is constrained to have positive values. The aimed factorization is in the form:

$$X \approx FG^T, \quad (3)$$

where we restrict G to be nonnegative, while placing no restriction on the signs of F . In the neighbor embedding case, we can reformulate (3) as $\mathbf{x}_t^i \approx X_d^i \mathbf{w}^i$, where the matrix to be approximated is a patch of the LR target image, the unconstrained factor is the matrix formed by its neighbors, and the nonnegative factor is the weight vector we want to compute.

In [7] an iterative solution to (3), that is proved to converge to a local minimum of the Euclidean distance $\|X - FG^T\|^2$, is found. The iterative solution consists of two multiplicative update rules, both for F and G elements. In our case, however, the matrix F , formed by the actual LR patches in the dictionary, is *fixed* and no update rule is needed for it. We implement then only the rule regarding the weight vector, which, with our notation, can be rewritten in the following way:

$$w_{ij} \leftarrow w_{ij} \sqrt{\frac{(\mathbf{x}_d^{jT} X_d^i)^+ + [\mathbf{w}^{iT} (X_d^{iT} X_d^i)]^-}{(\mathbf{x}_d^{jT} X_d^i)^- + [\mathbf{w}^{iT} (X_d^{iT} X_d^i)]^+}}, \quad (4)$$

where the positive and negative parts of a matrix A are defined respectively as $A_{ik}^+ = (|A_{ik}| + A_{ik})/2$ and $A_{ik}^- = (|A_{ik}| - A_{ik})/2$. The convergence of the formula, i.e. the non-increase of the approximation error $\|\mathbf{x}_t^i - X_d^i \mathbf{w}^i\|^2$ while updating only the weight vector, can be proved in the same way as in [7]. Thus, (4) can be used as an iterative formula to obtain non-negative weights for each patch combination.

3.2. Features and other parameters

As pointed out in [9], the hypothesis of similarity in the neighborhood structures between LR and HR patches, seen as vectors in two possibly distinct feature spaces, does not generally stand. In [3] a 4-value gradient representation (we call it “Feature 1”) is used for the LR patches; for the HR patches, instead, the mean-subtracted luminance values (we call this representation “Feature 2”) are used as features: after all HR patch feature vectors are recovered, the features are reversed by simply adding the average luminance value of the corresponding LR input patches. We keep the same scheme for the HR patches and test also *Feature 2* as a possible feature representation for LR patches. In fact, by using the same features for LR and HR patches, we think that we can better achieve the assumption of similarity on the neighborhood structures.

After choosing the method for computing the weights of the neighbor embedding and the features, we still have an important parameter to set in the neighbor embedding scheme:

the patch size. We experimentally found that the optimal patch size for a LR patch is 3×3 with a 2-pixel overlap. The size of the HR patches comes as a consequence of the magnification factor.

4. EXPERIMENTAL RESULTS

In order to make a comparison between the original LLE-based SR algorithm in [3] and our SNMF-based algorithm, we report the results for two images. In both cases, we start from an original HR image, which plays as the “ground-truth”, and we get the LR input image, by downsizing it. The external dictionaries, “relatively small” in size, are built from the same HR example images. Each example image is downsized by a factor equal to the magnification factor we want to achieve, and blurred by using a Gaussian filter, so obtaining a corresponding LR image. The patch pairs of the dictionaries are then obtained by cutting the LR and HR example images according to the patch sizes chosen (3×3 patches with a 2-pixel overlap in the LR case). The data on the dimensions of the two images are reported in Table 2. The dictionary size for the second test is larger, while using the same example images, because the LR example images were obtained by downsizing them by a smaller factor, so generating more patches.

Image	HR size	Factor	LR size	Dict. size
“head”	280×280	4	70×70	11844 pairs
“baby”	288×288	3	96×96	21168 pairs

Table 2: Dimensions of the two experimental images.

To evaluate the performance of the algorithm we used the *PSNR* (Peak Signal-to-Noise Ratio) as the measure of how close the super-resolved image is from the ground-truth. Fig. 1 and Fig. 2 show the results for the two methods for neighbor embedding, the one used in [3] (LS) and our SNMF-based method, in terms of *PSNR* against the number of neighbors (K), respectively using *Feature 1* (gradient representation) and *Feature 2* (luminance values mean-subtracted) to create the LR patch feature vectors.

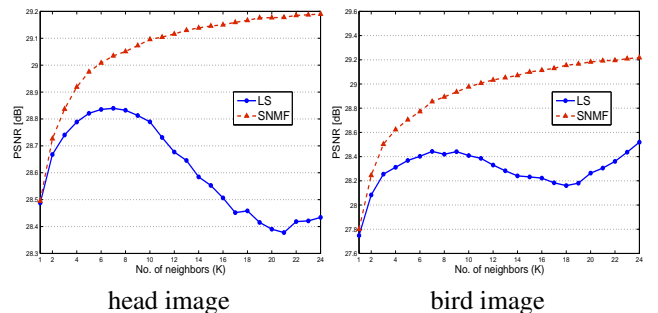


Fig. 1: Comparison between LS and SNMF (*PSNR* against the number of neighbor) using *Feature 1*.

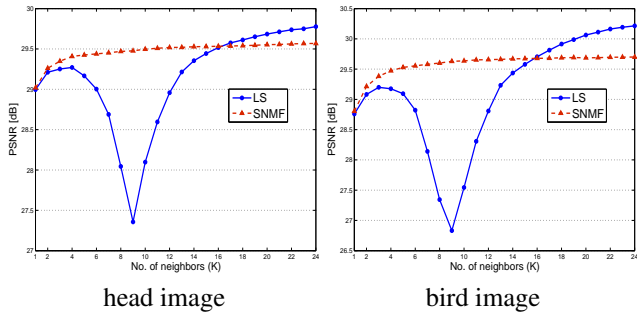


Fig. 2: Comparison between LS and SNMF (*PSNR* against the number of neighbor) using *Feature 2*.

As we can see from the figures, SNMF behaves much more regularly than LS, with a slightly increasing curve in both cases, and generally outperforms it. On the other hand, the behavior of LS is diametrically opposite, depending on which feature we choose. In the case of gradient features (*Feature 1*), the *PSNR* values reach a maximum and then rapidly decreases; for *Feature 2*, instead, we have a fall in the performance for medium-range values of neighbors selected and then the *PSNR* starts to rise again. We can also see that with *Feature 2* we have higher *PSNR* values.

In Fig. 3 some visual results are showed on the comparison between bicubic interpolation, the LLE-based Neighbor Embedding algorithm and our SNMF-based algorithm.

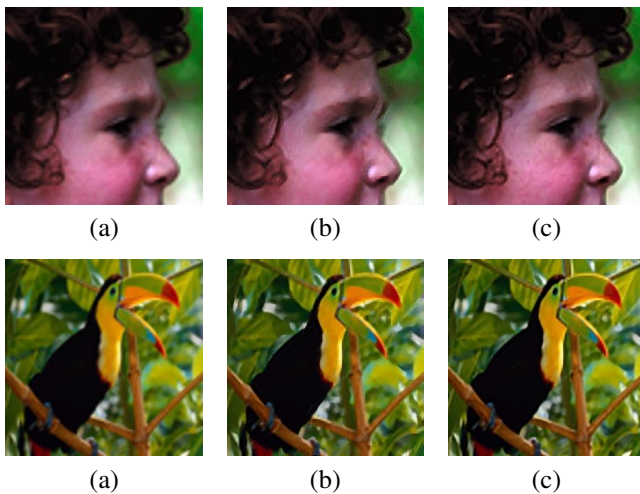


Fig. 3: Visual comparison on the head and bird images between simple bicubic interpolation (a), the LLE-based Neighbor Embedding algorithm (b) in a “standard” configuration (*Feature 1*, $K = 5$), and our SNMF-based algorithm (c) with *Feature 2* and $K = 10$.

5. CONCLUSION

In this paper we described a new method for single-image SR, that uses an external dictionary and is based on the neighbor embedding approach. With respect to the LLE-based algo-

rithm present in the literature, we proposed a different way to embed the neighbors of each patch, so that the corresponding weight coefficients turn out to be non-negative. We derived it from an algorithm designed to perform semi-nonnegative matrix factorizations (SNMF), i.e. with only one factor constrained to have all positive values. Moreover, we used centered pixel values as the features to represent each patch (both LR and HR), differently from the LLE-based algorithm where a gradient-based representation is used for LR patches. We showed that our new SNMF-based SR algorithm has more regular performance as the number of neighbors varies, and generally higher *PSNR* values on the output image. In particular, by using the new features and choosing a number of neighbors relatively large (e.g. $K \geq 10$) we have constantly good results. Visual results also confirm this evaluation.

6. REFERENCES

- [1] William T. Freeman, Thouis R. Jones, and Egon C. Pasztor, “Example-Based Super-Resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
- [2] Kwang In Kim and Younghee Kwon, “Example-Based Learning for Single-Image Super-Resolution,” in *Pattern Recognition: 30th DAGM Symposium*. 2008, vol. 5096 of *Lecture Notes in Computer Science*, pp. 456–465, Springer.
- [3] Hong Chang, Dit-Yan Yeung, and Yimin Xiong, “Super-Resolution Through Neighbor Embedding,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, vol. 1, pp. 275–282.
- [4] Jianchao Yang, J. Wright, T. Huang, and Yi Ma, “Image Super-Resolution as Sparse Representation of Raw Image Patches,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.
- [5] Jinjun Wang, Shenghuo Zhu, and Yihong Gong, “Resolution enhancement based on learning the sparse association of image patches,” *Pattern Recognition Letters*, vol. 31, pp. 1–10, 1 2010.
- [6] Sam T. Roweis and Lawrence K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [7] Chris Ding, Tao Li, and Michael I. Jordan, “Convex and Semi-Nonnegative Matrix Factorizations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [8] Daniel D. Lee and H. Sebastian Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [9] K. Su, Q. Tian, Q. Xue, N. Sebe, and J. Ma, “Neighborhood issue in Single-frame Image Super-Resolution,” in *2005 IEEE International Conference on Multimedia and Expo (ICME)*, 2005, pp. 1122–1125.