Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding

Marco Bevilacqua[1]
marco.bevilacqua@inria.fr
Aline Roumy[1]
aline.roumy@inria.fr
Christine Guillemot[1]
christine.guillemot@inria.fr
Marie-Line Alberi Morel[2]
marie_line.alberi-morel@alcatel-lucent.com

[1] INRIA
Campus universitaire de Beaulieu,
35042 Rennes Cedex, France

[2] Alcatel-Lucent, Bell Labs France
Route de Villejust,
91620 Nozay, France

Abstract

This paper describes a single-image super-resolution (SR) algorithm based on nonnegative neighbor embedding. It belongs to the family of single-image example-based SR algorithms, since it uses a dictionary of low resolution (LR) and high resolution (HR) trained patch pairs to infer the unknown HR details. Each LR feature vector in the input image is expressed as the weighted combination of its K nearest neighbors in the dictionary; the corresponding HR feature vector is reconstructed under the assumption that the local LR embedding is preserved. Three key aspects are introduced in order to build a low-complexity competitive algorithm: (i) a compact but efficient representation of the patches (feature representation) (ii) an accurate estimation of the patches by their nearest neighbors (weight computation) (iii) a compact and already built (therefore external) dictionary, which allows a one-step upscaling. The neighbor embedding SR algorithm so designed is shown to give good visual results, comparable to other state-of-the-art methods, while presenting an appreciable reduction of the computational time.

1 Introduction

Single-image super-resolution (SR) refers to a family of techniques that map an input low resolution (LR) image into a single high resolution (HR) image. Depending on the way this mapping is obtained, SR techniques can be broadly classified into two main approaches: inverse problem methods (e.g. [4, 6, 10]), where SR is seen as an ill-posed problem and regularization tools are used to solve it, and machine learning (ML) methods, that use a dictionary and aim at learning the mapping.

In ML techniques, as said, the mapping from the LR image to the HR image is learned by using a dictionary. *Example-based SR* algorithms belong to this family: the learning process is performed locally, by trying to infer the local HR details through the use of small "examples". For general SR purposes the examples used consist of patches (sub-windows

of image) and the dictionary is therefore formed by pairs of LR and HR patches. Example-based SR algorithms can differ in the number of candidates taken from the dictionary and used as predictors (single or several candidates) and the way the patches are combined in order to generate the HR outputs. Another difference is given by the nature of the dictionary: external (derived from a set of training images) or internal (the patch correspondences are derived from the image itself, exploiting self-similarities, e.g. [7, 8]).

Example-based SR algorithms can further be divided into two approaches. In *local learning* methods, the mapping from a LR patch to a HR patch is inferred by using Support Vector Regression [11], linear or Kernel Ridge Regression [13]. The other approach is *Neighbor Embedding (NE)* [1, 2, 5], that first learns the local neighborhood of a LR patch and assumes that this is the best estimate for the local neighborhood of the corresponding HR patch. More precisely, the basic assumption is that the LR and HR patches lie on manifolds with similar local geometries: as a consequence of that, once the LR input patch is expressed as the linear combination of a certain number of its neighbors taken from the dictionary, the HR 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. The way to compute the local neighbor embedding is derived from a manifold learning method for dimensionality reduction called locally linear embedding (LLE) [12]. LR and HR patches are expressed as "feature vectors" and the whole SR procedure is carried out in the feature space.

In the wake of these NE-based algorithms, we propose a new single-image SR algorithm, that achieves better performance and a low complexity target. With respect to the LLE-based weight computation method, we propose a new method, based on a Least Squares (LS) approximation of the LR patches with a non-negativity constraint, and prove the effectiveness of the choice of having non-negative weights. Moreover, we discuss the issue of the feature representation and propose to use centered luminance values as unique "low-cost" features for both LR and HR patches. As the low-complexity is one of the targets of the algorithm, we focus on a one-step procedure that directly produces the desired upscaled image, showing that, for the sake of this choice, using an external dictionary is preferable.

The rest of the paper is organized as follows. Section 2 briefly describes the NE-based approach to SR. Then, in Section 3, we provide the details of our algorithm, motivating the choice of the feature representation and the weight computation method adopted: a performance analysis and a complete description of the final procedure are given. In Section 4 some visual results are showed. Finally, Section 5 gives some concluding remarks and outlines for future work.

2 Neighbor embedding approach to super-resolution

In the NE approach we have a dictionary $\mathcal{D} = (\mathcal{X}_d, \mathcal{Y}_d)$, formed by several LR/HR patch co-occurrences, where $\mathcal{X}_d = \{\mathbf{x}_d^j\}_{j=1}^{N_d}$ is a set of LR patch vectors and $\mathcal{Y}_d = \{\mathbf{y}_d^j\}_{j=1}^{N_d}$ is the set related to their HR counterparts. The basic idea is that we can express a LR input patch as the weighted combination of its LR K nearest neighbors (K-NN) selected from the dictionary, and then apply the same weighted combination to the corresponding HR patches in the dictionary to reconstruct the HR output patch. All patches, both from the dictionary and the target images, are transformed into *feature vectors*, by concatenating some features computed on the pixels of the patches. For convenience, in order not to introduce other notations, in this paper all the vectors representing patches are intended as their related feature vectors. The first step is to divide the LR input image X into patches of the same size of the

LR patches in the dictionary, and convert them into feature vectors, so obtaining the set of LR feature vectors $\mathcal{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{N_t}$. The algorithm then proceeds as follows:

1. For each LR patch feature vector $\mathbf{x}_t^i \in \mathcal{X}_t$
   (a) Find its K-NN in $\mathcal{X}_d$ in terms of Euclidean distance:
   $$\mathcal{N}_i = \underset{\mathbf{x}_d^j \in \mathcal{X}_d}{\arg\min^K} \left\| \mathbf{x}_t^i - \mathbf{x}_d^j \right\|^2 . \quad (1)$$
   (b) Find a weighted combination that approximates $\mathbf{x}_t^i$ with the selected neighbors, i.e. compute the K weights $\{w_{ij}\}_{j=1}^K$ such that:
   $$\mathbf{x}_t^i \approx \sum_{\mathbf{x}_d^j \in \mathcal{N}_i} w_{ij} \mathbf{x}_d^j . \quad (2)$$
   (c) Apply the same weights for the reconstruction of the output HR patch feature vector $\mathbf{y}_t^i$ with the corresponding neighbors in $\mathcal{Y}_d$:
   $$\mathbf{y}_t^i = \sum_{\mathbf{y}_d^j \in \mathcal{H}(\mathcal{N}_i)} w_{ij} \mathbf{y}_d^j . \quad (3)$$
   where $\mathcal{H}(\mathcal{N}_i)$ indicates the set of HR feature vectors in the dictionary corresponding to the LR neighborhood $\mathcal{N}_i$.
2. Once all the HR patch feature vectors are generated, reverse them back to pixel-based patches, and combine the obtained patches to form the output image.

In previous NE-based SR algorithms [1, 2, 5], the weights of each linear combination (Step 1b) are computed by minimizing the approximation error of the related LR patch, $\varepsilon^i = \|\mathbf{x}_t^i - \sum_{\mathbf{x}_d^j \in \mathcal{N}_i} w_{ij} \mathbf{x}_d^j\|^2 = \|\mathbf{x}_t^i - X_d^i \mathbf{w}^i\|^2$, subject to the condition that they sum up to one. Namely, they are the result of the following constrained least squares (LS) minimization problem (*SUM1-LS*):

$$\mathbf{w}^i = \underset{\mathbf{w}}{\arg\min} \|\mathbf{x}_t^i - X_d^i \mathbf{w}\|^2 \quad \text{s.t.} \quad \mathbf{1}^T \mathbf{w} = 1 . \quad (4)$$

A solution to (4) can be found through the method of Lagrange multipliers.

In the following section we propose an alternative criterion to compute the weights of the neighbor embedding and study the issue of the feature representation. This will lead us to the formulation of a new NE-based SR algorithm.

3 Proposed algorithm

3.1 Feature representation

As mentioned before, the entire neighbor embedding SR procedure is carried out in a feature space: LR and HR patches are represented by vectors of features, namely some transformations of the luminance values of the pixels of the patch. The role of the features is double: 1) to catch the most relevant part of the LR information in order to have a good "predictor" for the HR patch reconstruction; 2) to possibly enforce the similarity between the LR and HR patch structures.

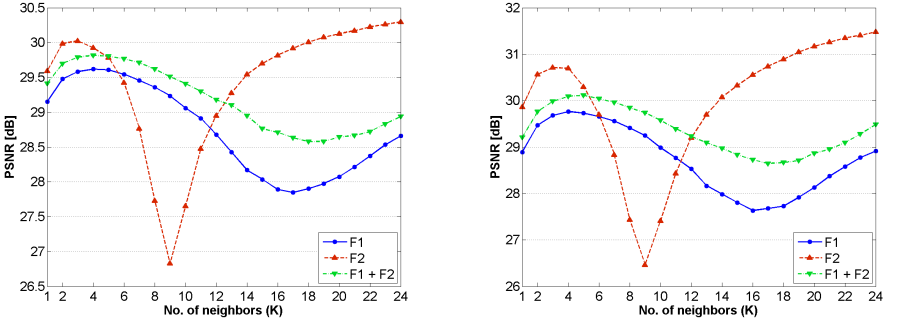
In order to pursue the first purpose, several algorithms in the literature propose to preprocess the LR images according to the idea that the middle and high-frequency content of

the LR patches is the most helpful information to learn the LR-HR patch correspondences. In the original NE-based algorithm [2], e.g. simple features derived from the first and second-order gradient are taken into account. As for the representation of the HR training patches, the common solution in the literature is to straightly use the luminance values of the HR training images, possibly after doing an operation of contrast normalization or mean removal. In our implementation of the neighbor embedding approach, we test three simple feature configurations for the LR patches.

- F 1 *First order gradient*: first-order derivatives in both directions are computed for each pixel of the patch (2 values per pixel), by applying to the luminance matrix $g_1 = [-1,0,1]$ and $g_2 = g_1^T$ as 1-D filters. Let N be the number of pixels in a patch and d the dimension of the LR feature vectors, then $d = 2N$.
- F 2 *Centered luminance values*: the features are obtained by taking the luminance values and subtracting the mean value. This corresponds to remove the DC component of the patch; we can therefore see also this method as providing a low-cut filtered version of the LR patch. In this case, $d = N$.
- F 3 *F1+F2*: concatenation of F1 and F2 (thus, $d = 3N$) as considered as in [1].

In order to pursue the second purpose, we would like to keep the same feature space for the LR and HR patches. However, at the end of the SR algorithm we have to reverse the features back to pixel-based patches. This reversion step is not feasible in the case of gradient feature vectors (more "unconstrained" equations than unknowns); we can reverse the centered features, instead, by simply adding the mean value of the corresponding LR patches in the input image. Therefore, F2 is the only possible choice for the the HR features.

Among the 3 possible options for the LR features (F1, F2, F1+F2), we expect centered luminance values (F2) to be the most intuitive solution, as the representation for LR and HR patches would be unique. Moreover, this is the "cheapest" solution in terms of computational time, as the feature vectors contain less entries than for the other feature representations. In [1] a performance analysis of the feature representations is given, including gradient features and centered luminance values ("norm luminance"): the authors conclude that the best solution for the LR features is a weighted concatenation of norm luminance and first-order gradient values, while purely centered luminance values representing the fourth or fifth choice (see Figure 3 in [1]). However, this analysis is not performed by considering variations of K (number of neighbors). Figure 1 evaluates for two test images ("head" and "bird") the performance of the algorithm (*PSNR* of the super-resolved image) versus K, for the chosen LR features and the standard *SUM1-LS* method used to compute the weights.



head bird

Figure 1: Comparison between LR feature representations, *SUM1-LS* used as NE method.

Figure 1 shows that the performance of the algorithm is highly dependent on the number of neighbors K. For all the feature representations, we observe that the curves present an "up-down-up" behavior; the fall in the case of centered features appears even dramatic. We explain this behavior with the "blindness" of the NE approach: the weights of the patch combinations are computed on the LR feature vectors and then blindly applied to the corresponding HR vectors. As we observe, a problem arises at a certain critical point that depends on d. For F2, we can make the following observations.

Observation 3.1. *Let d be the dimension of the LR vectors, K the number of neighbors, and r_i the rank of the ith neighborhood matrix X_d^i. Then, the neighbors (centered) vectors lie on a $d-1$-dimensional hyperplane, and the rank of the neighborhood matrix is upper-bounded as follows: $r_i \leq min(d-1,K)$. For the dictionary built as in Section 3.3, we observe that the upper bound is tight. More precisely, $r_i = min(d-1,K)$ with high probability.*

Observation 3.2. *For $K = d$ the SUM1-LS problem is assimilable to the solution of a square linear system, as we have d equations (the $d-1$ linearly independent equations of $\mathbf{x}_t^i \approx X_d^i \mathbf{w}^i$ plus the one given by the equality constraint) in $K = d$ unknowns. Here, experimentally we have a "critical point" in the performance.*

Intuitively, we explain this criticality with the fact that the LS solution is "too fitted" on the LR data and thus generates undesired bad HR reconstructions. In Figure 1 this point is, in case of F2, at $K = d = 9$, as we use 3×3 LR patches.

Nevertheless, we observe that outside the fall region, centered luminance features outperform the other feature representations, thus showing the expected potential. Therefore, we decide to use F2 as unique features for LR and HR patches. In the following section we propose a new criterion for the weight computation, in order to avoid the irregular performance with K and fully exploit the common features.

3.2 Non-negative embedding

In the previous section we pointed out that, for any feature chosen, we have a fall of the performance for a certain critical point, by using *SUM1-LS* as a NE method.

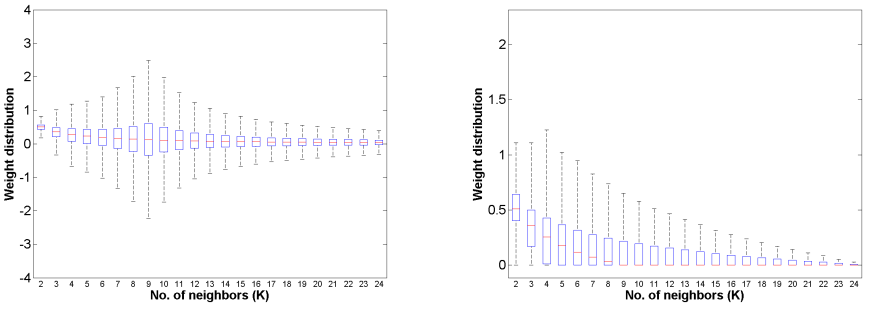
We believe that this problem can be avoided by replacing the sum-to-one equality constraint by a more "relaxed" inequality constraint. Therefore, we propose another method for computing the neighbor embedding, derived from a LS problem equivalent to (4), but with a non-negativity inequality constraint, according to "the intuitive notion of combining parts to form a whole" [9] (only additive combinations of patches are allowed):

$$\mathbf{w}^i = \arg\min_{\mathbf{w}} \|\mathbf{x}_t^i - X_d^i \mathbf{w}\|^2 \quad \text{s.t.} \quad \mathbf{w} \geq 0\,. \tag{5}$$

The problem in (5) is known in the literature as non-negative least squares (NNLS); an iterative solution is provided in [3, Ch. 23, p. 161]. The formula experimentally turns out to converge after an average of 2 iterations.

In Figure 2 the weight distributions of the two NE criteria are compared, using the box plot representation. Each box is delimited by the 25th (Q1) and 75th percentile (Q3). Values that exceed the boxes by $2 \times (Q3 - Q1)$ are considered outliers and not drawn. Interestingly, the values for *SUM1-LS* weights get larger in the proximity of the critical points; instead, the values of the *NNLS* weights regularly decrease with K.

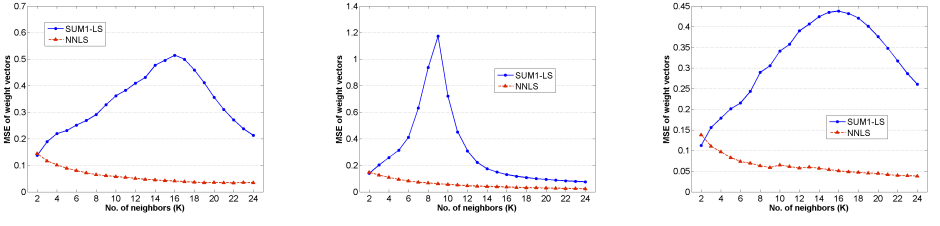
To further evaluate the goodness of the new nonnegative NE criterion, compared to *SUM1-LS*, we used the distance (in terms of MSE) between the LR weights, resulted by



SUM1-LS NNLS

Figure 2: Distribution of the weights for each value of K; F2 used as LR features.

the approximation of a LR test vector with its LR neighbors, and the HR weights, obtained by approximating the corresponding HR test vector with the HR vectors related to the LR neighbors found. This is an index of how the LR weights, what we can actually compute, reflect the ideal HR weights. We averaged the error over 1000 randomly selected test vectors (Figure 3).



F1 F2 F1 + F2

Figure 3: Comparison between *SUM1-LS* and *NNLS* in terms of MSE of the LR/HR weight vectors.

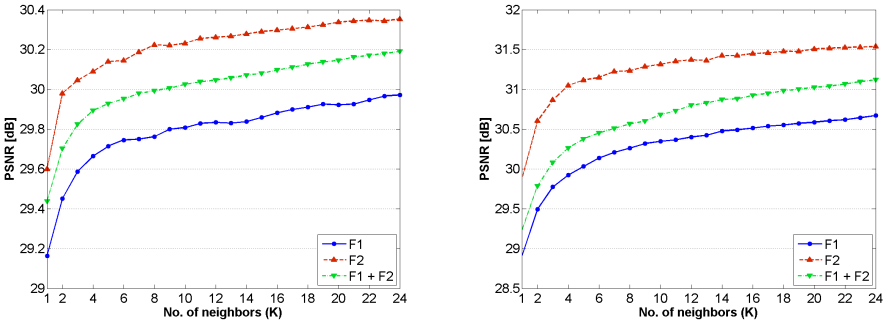
Figure 3 shows that the new nonnegative embedding method is much more suitable for computing the weights of a common embedding, as the MSE error between the weight vectors is, in any case, below the corresponding one for *SUM1-lS*. The results are confirmed in Figure 4, where a comparison between different features (in terms of ouput *PSNR*), while using *NNLS*, is presented.

Compared to Figure 1, Figure 4 presents a much more stable performance, as K varies: the *PSNR* curve is monotonically increasing. Moreover, the *PSNR* values are generally higher and F2 are clearly the winning features.

3.3 Dictionary design

From the analysis undertaken in the previous sections, we can derive a new feature-based NE-based SR algorithm: centered luminance values (F2) are taken as features for both LR and HR patches, and the new NE method with nonnegative weights is chosen to compute the patch combinations.

An important issue for example-based SR algorithms is represented by the choice of the dictionary. To pursue the aim of realizing a low-complexity algorithm, we decide to adopt



head bird

Figure 4: Comparison between LR feature representations, *NNLS* used as NE method.

		Internal DB		Ext DB "esa"		Ext DB "wiki"	
Image	**Magn. Factor**	*PSNR*	DB size	*PSNR*	DB size	*PSNR*	DB size
Baby	4	27.95	2179	30.62	56514	30.32	218466
Bird	3	27.72	2256	31.37	100966	31.42	389232
Woman	2	27.29	15044	30.91	229096	30.44	880440

Table 1: Final *PSNR* and DB size (i.e. number of pairs of patches) for different dictionaries.

a one-step upscaling procedure, whereas other algorithms (e.g. [7, 8]) achieve the desired magnification factor by several upscalings with smaller scale factors (the SR procedure is thus iterated several times). For the dictionary, we have then two possibilities: 1) build an external dictionary from a set of training images (from the original HR images, generate the LR versions, and extract HR and LR patches, respectively); 2) learn the patch correspondences in a pyramid of recursively scaled images, starting from the LR input image, in the way of [8]. In Table 1 results from the pyramid internal dictionary and two external dictionaries are reported; *NNLS* with $K = 12$ is used as the NE method. For both the external dictionaries, a small number of natural images is taken. As we can see from the table, the external dictionary performs significantly better, since the number of pairs of patches we can derive from the internal pyramid is insufficient. On the other hand, the size of the external dictionary can be tuned to any value, by conveniently choosing the training images. In this paper, we choose a dictionary s.t. the upscaling of a 70×70 image by a factor of 4 takes about 5 seconds.

There is one important parameter left to set in the algorithm: the size of the LR patches taken from the LR input image. From our experiments we found that the optimal size is 3×3 with a 2-pixel overlap. As we perform a one-step upscaling, the size and the overlap degree of the HR patches come as consequences of the magnification factor: e.g. with a factor of 4 we will have 12×12 HR patches with a 8-pixel overlap. The reason for choosing largely overlapping HR patches is due to the fact that at the end of the NE algorithm (see Step 2 of the general procedure in Section 2) we combine the patches together by simply averaging the related pixel values in the overlapping regions. By having big overlaps, we can in fact implicitly impose smooth properties on the output image. Moreover, as an additional tool used by several SR algorithms (e.g. [8]), we implement a final check to assure the super-resolved image to be consistent with the LR input image: an *iterative back projection* (IBP). Note that [8] uses IBP, that stops on average after 3 iterations, at each step; we use it only once instead.

4 Results

In this section some visual results and comparisons with other methods are presented. In particular, our algorithm is compared to the original LLE-based NE algorithm of Chang et al. [2], to [8] (using a third party implementation), which is currently considered among the most advanced single-image SR algorithms, and to the KRR method of Tang et. al [13]. Table 2 summarizes the results for several images and magnification factors, reporting the output *PSNR* and the running time in seconds as an indication of the complexity of the algorithm. Figure 5 offers a few visual results for two images; more results can be found in http://www.irisa.fr/prive/Aline.Roumy/results/SR_BMVC12.html.

When comparing our algorithm to Chang et al. [2] the visual improvements are evident: our algorithm is able to super-resolve finer details, whereas the results of [2] are often affected by ringing artifacts. The impression is confirmed by reading the *PSNR* values. With respect to the local learning method of Tang et al. [13], too, the PSNR achieved by our algorithm are always higher. As for the running time, this turns to be in favor of our algorithm in both comparisons, thanks to the low-complexity choice of the features (F2 instead of gradient features). The comparison with the method of Glasner et al. [8] is also satisfying: [8] gives generally higher values of *PSNR*, although our method is better performing for a magnification factor equal to 2. Nevertheless, the visual results are fairly comparable. What represents an issue for [8] is the complexity, as the algorithms involves several steps and the dictionary is iteratively updated by taking patches from the image "pyramid": although the values provided only serve to give an idea of the complexity, the algorithm clearly requires much more time than ours and the running time grows exponentially with the size of the input image.

		Our algorithm		Chang et al.		Glasner et al.		Tang et al.		Lin. Regr. + F2	
Image	*Scale*	*PSNR*	Time	*PSNR*	Time	*PSNR*	Time	*PSNR*	Time	*PSNR*	Time
baby	2	34.64	58	33.42	339	34.66	4083	33.72	425	34.76	30
bird	2	34.69	18	32.94	110	34.42	406	33.31	132	34.91	9
butterfly	2	27.54	17	25.90	77	26.83	265	26.05	82	27.66	7
head	2	32.88	18	32.34	145	32.68	367	32.43	151	32.88	8
woman	2	30.91	15	29.43	114	30.61	410	29.64	128	31.01	8
baby	3	32.44	27	31.00	116	32.94	2188	31.47	111	32.59	12
bird	3	31.37	9	29.71	47	32.16	281	30.07	42	31.57	5
butterfly	3	24.31	9	22.58	34	25.66	232	22.72	25	24.47	4
head	3	31.46	12	30.82	68	31.69	370	30.95	54	31.55	4
woman	3	27.98	12	26.45	37	28.79	248	26.66	37	28.12	4
baby	4	30.62	22	29.27	86	31.41	4381	29.70	81	30.76	13
bird	4	28.99	6	27.37	21	30.07	475	27.84	22	29.12	3
butterfly	4	22.05	7	20.50	18	23.94	315	20.61	13	22.06	3
head	4	30.26	6	29.57	26	30.86	379	29.83	28	30.43	3
woman	4	25.66	5	24.25	17	26.79	401	24.46	20	25.69	3

Table 2: Results (*PSNR* and running time in sec.) for different images.

5 Conclusion and future work

In this paper we proposed a novel algorithm for single-image SR, based on neighbor embedding. With the double target of low complexity and good performance, we handled with three aspects: the representation of the patches, the weight computation method, and the dictionary design. We chose to use centered luminance values as unique "low-cost" solution

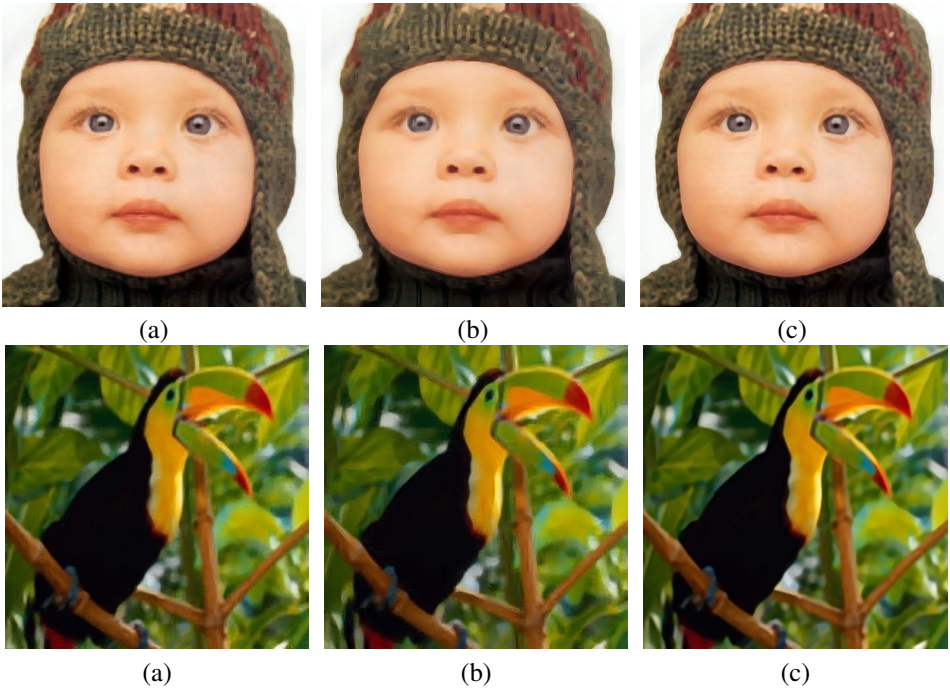


Figure 5: Results for the "baby" ($\times 4$) and the "bird" ($\times 3$) images: (a) Our algorithm - (b) Chang et al. [2] - (c) Glasner et al. [8] (7 steps for "baby", 5 steps for "bird").

for both the LR and HR feature-based patch representations, and proposed a new method for computing the weights based on nonnegative LS. We chose, then, to use an external dictionary, showing that this solution is more convenient than having an internal dictionary, when a single-step magnification is performed. The algorithm so designed, when compared to other single-image SR methods, is shown to give good visual results and to present a much reduced time complexity, thanks to the one-step upscaling procedure adopted and the feature representation chosen. As for the future work, we believe that the NE-based procedure can be improved in the neighbor selection step: different metrics than the Euclidean distance with the LR candidate can be taken into account in order to select neighbors more suitable for the HR reconstruction. In the flavor of [13], we also plan to investigate methods based on the local learning of the LR-HR mapping function, and to consider in this case too the issue of the features. In fact, as we can see from the last column of Table 2, by applying our features (centered luminance values) to a simple linear regression model, we already obtain promising results.

References

[1] Tak-Ming Chan, Junping Zhang, Jian Pu, and Hua Huang. Neighbor embedding based super-resolution algorithm through edge detection and feature selection. *Pattern Recognition Letters*, 30(5):494–502, 4 2009. ISSN 0167-8655.

[2] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-Resolution Through Neighbor

Embedding. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 275–282, 2004.

[3] Richard J. Hanson Charles L. Lawson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[4] Shengyang Dai, Mei Han, Wei Xu, Ying Wu, Yihong Gong, and A.K. Katsaggelos. SoftCuts: A Soft Edge Smoothness Prior for Color Image Super-Resolution. *IEEE Transactions on Image Processing*, 18(5):969–981, 5 2009.

[5] Wei Fan and Dit-Yan Yeung. Image Hallucination Using Neighbor Embedding over Visual Primitive Manifolds. In *2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 7 2007.

[6] Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and Robust Multi-Frame Super-Resolution. *IEEE Transactions on Image Processing*, 13(10): 1327–1344, 10 2004.

[7] Gilad Freedman and Raanan Fattal. Image and Video Upscaling from Local Self-Examples. *ACM Trans. Graph.*, 28(3):1–10, 2010. ISSN 0730-0301.

[8] Daniel Glasner, Shai Bagon, and Michal Irani. Super-Resolution from a Single Image. In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pages 349–356, 10 2009.

[9] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 10 1999.

[10] Ali Mohammad-Djafari. Super-Resolution: A Short Review, A New Method Based on Hidden Markov Modeling of HR Image and Future Challenges. *The Computer Journal*, 52(1):126–141, 2008.

[11] K.S. Ni and T.Q. Nguyen. Image Superresolution Using Support Vector Regression. *IEEE Transactions on Image Processing*, 16(6):1596–1610, 6 2007. ISSN 1057-7149.

[12] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 12 2000.

[13] Yi Tang, Pingkun Yan, Yuan Yuan, and Xuelong Li. Single-image super-resolution via local learning. *International Journal of Machine Learning and Cybernetics*, 2:15–23, 2011.