FAST SINGLE-IMAGE SUPER-RESOLUTION WITH FILTER SELECTION

Jordi Salvador *Eduardo Pérez-Pellitero* *Axel Kochale*

Image Processing Lab
Technicolor R&I Hannover

ABSTRACT

This paper presents a new method for estimating a super-resolved version of an observed image by exploiting cross-scale self-similarity. We extend prior work on single-image super-resolution by introducing an adaptive selection of the best fitting upscaling and analysis filters for example learning. This selection is based on local error measurements obtained by using each filter with every image patch, and contrasts with the common approach of a constant metric in both dictionary-based and internal learning super-resolution. The proposed method is suitable for interactive applications, offering low computational load and a parallelizable design that allows straight-forward GPU implementations. Experimental results also show how our method generalizes better to different datasets than dictionary-based super-resolution and comparably to internal learning with adaptive post-processing.

Index Terms— Super-resolution, Raised cosine, Cross-scale self-similarity, Parallel algorithms

1. INTRODUCTION

First efforts in Super-Resolution (SR) focused on classical multi-image reconstruction-based techniques [1, 2]. In this approach, different observations of the same scene captured with sub-pixel displacements are combined to generate a super-resolved image. This constrains the applicability to very simple types of motion between captured images, since registration needs to be done, and it is typically unsuitable for upscaling frames in most video sequences. It also degrades fast whenever the magnification factor is large [3, 4] or the number of available images is insufficient.

The SR research community has overcome some of these limitations by exploring the so called Single-Image Super Resolution (SISR). This alternative provides many possible solutions to the ill-posed problem of estimating a high-resolution (HR) version of a single input low-resolution (LR) image by introducing different kinds of prior information. One common approach in SISR is based on machine learning techniques, which aim to *learn* the relation between LR and HR images, usually at a patch level, using a training set of HR images from which the LR versions are computed [5, 6, 7]. Thus, performance will be closely related to the content of the training information. To increase the generalization capability we need to enlarge the training set, resulting in a growing computational cost. If we consider all possible image scenarios (e.g. ranging from animals to circuitry), finding a generalizable training set can then be unfeasible. Current research on sparse representation [8] tackles this problem by representing image patches as a sparse linear combination of base patches from an optimal over-complete dictionary. Even though with sparse representation the dictionary size is drastically reduced and so the querying times, the execution time of the whole method is still lengthy, as observed in Section 3. In addition, the cost of finding the sparse representation (which is not taken into account in our tests) is still conditioned by the size of the training dataset, thus there might still be generalization issues.

There also exist methods with *internal learning* (i.e. the patch correspondences/examples are obtained from the input image itself), which exploit the *cross-scale self-similarity* property [9, 10]. The method we present in this paper follows this strategy, aiming at a better execution time vs. quality trade-off. In Section 2 we present the fundamental mechanism for internal learning we use in our method, followed by our adaptive filter selection, which leads to better generalization to the non-stationary statistics of real-world images.

In Section 3 we show quantitative results (PSNR, SSIM and execution time) obtained with different datasets, as well as qualitative evidence that support the validity of the proposed approach in comparison to two state-of-the-art SISR methods. These results show that our method 1) is orders of magnitude faster than the compared SISR methods; and 2) the visual quality of the super-resolved images is comparable to that of the internal learning SISR method [11] and slightly superior than that of the dictionary-based one [8], being the latter affected by the problem of limited generalization capability.

2. PROPOSED METHOD

When using interpolation-based upscaling (e.g. bicubic or bilinear) methods, the resulting HR image presents a frequency spectrum with shrunk support. Interpolation cannot fill-in the missing high-frequency band up to the wider Nyquist limit for the upscaled image. In our method, the high-frequency

band is estimated by combining high-frequency examples extracted from the input image and added to the interpolated low-frequency band, based on a similar mechanism to the ones used by [12] (targetting *demosaicking*) or [13] (SISR).

As originally presented in [9], most images present the *cross-scale self-similarity* property. This basically results in a high probability of finding very similar patches across different scales of the same image. Let $\mathbf{x}_l = \mathbf{h}_s * (\mathbf{y} \uparrow s)$ be an upscaled version of the input image $\mathbf{y}$, with $\mathbf{h}_s$ a linear interpolation kernel and s the upscaling factor. The subscript l refers to the fact this upscaled image only contains the low-frequency band of the spectrum (with normalized bandwidth $1/s$). We just assume $\mathbf{h}_s$ has a low-pass behavior, but more details about the filter are given in Section 2.1.

The input image $\mathbf{y}$ can be analyzed in two separate bands by using the same interpolation kernel used for upscaling. We can compute its low-frequency $\mathbf{y}_l = \mathbf{h}_s * \mathbf{y}$ and high-frequency $\mathbf{y}_h = \mathbf{y} - \mathbf{y}_l$ bands. By doing so, we are generating pairs of low-frequency references (in $\mathbf{y}_l$) and their corresponding high-frequency examples (in $\mathbf{y}_h$). We should note that $\mathbf{y}_l$ has the same normalized bandwidth as $\mathbf{x}_l$ and, most importantly, the cross-scale self-similarity property is also present between these two images.

Let $\mathbf{x}_{l,i}$ be a patch with dimensions $N_p \times N_p$ pixels with the central pixel in a location $\lambda(\mathbf{x}_{l,i}) = (r_i, c_i)$ within $\mathbf{x}_l$. We look for the best matching patch in the low-resolution low-frequency band $\mathbf{y}_{l,j} = \arg\min_{\mathbf{y}_{l,j}} \|\mathbf{y}_{l,j} - \mathbf{x}_{l,i}\|_1$, whose location is $\lambda(\mathbf{y}_{l,j})$ [1]. This is also the location of the high-frequency example $\mathbf{y}_{h,j}$ corresponding to the low-frequency patch of minimal cost. This search is constrained to a window of size $N_w \times N_w$ pixels around $\lambda(\mathbf{x}_{l,i})/s$, assuming it is more likely to find a suitable example in a location close to the original one than further away [13].

The local estimate of the high-frequency band corresponding to a patch $\mathbf{x}_{l,i}$ is just $\mathbf{x}_{h,i} = \mathbf{y}_{h,j}$. However, in order to ensure continuity and also to reduce the contribution of inconsistent high-frequency examples, the patch selection is done with a sliding window, which means up to $N_p \times N_p$ high-frequency estimates are available for each pixel location λ_i. Let $\mathbf{e}_i$ be a vector with these $n \leq N_p \times N_p$ high-frequency examples and $\mathbf{1}$ an all-ones vector. We can find the estimated high-frequency pixel as $x_i = \arg\min_{x_i} \|\mathbf{e}_i - x_i\mathbf{1}\|_2^2$, which results in $x_i = \sum_{j=1}^{n} e_{i,j}/n$, although different norms might also be considered.

Once the procedure above is applied for each pixel in the upscaled image, the resulting high-frequency band $\mathbf{x}_h$ might contain low-frequency spectral components since 1) filters are not ideal and 2) the operations leading to $\mathbf{x}_h$ are non-linear. Thus, in order to improve the spectral compatibility between $\mathbf{x}_l$ and $\mathbf{x}_h$, we subtract the low-frequency spectral component from $\mathbf{x}_h$ before we add it to the low-frequency band to generate the reconstructed image $\mathbf{x} := \mathbf{x}_l + \mathbf{x}_h - \mathbf{h}_s * \mathbf{x}_h$.

[1] $\|\mathbf{x}\|_P = \left(\sum_{i=1}^{n} |x_i|^P\right)^{1/P}$ is the P-norm of a patch with n pixels

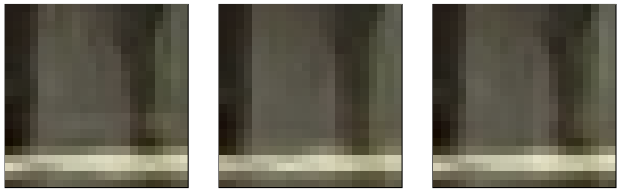


Fig. 1. Effects of filter ($\mathbf{h}_s$) selection ($2\times$ magnification). In (a), a very selective filter provides detailed texture in the super-resolved image but also produces ringing. In (b), a filter with small selectivity reduces ringing but fails to reconstruct texture. In (c), texture is reconstructed with reduced ringing by locally selecting a suitable filter.

2.1. Filter selection

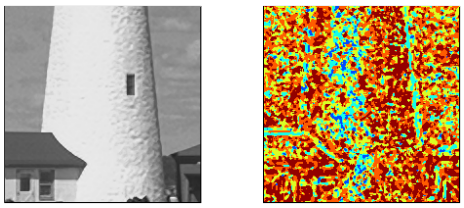
In Fig. 1 (a) and (b) we show how the proposed method behaves when considering different designs for the interpolation kernel (or low-pass filters) $\mathbf{h}_s$. Overall, the choice of a selective filter provides a good texture reconstruction in the super-resolved image, whereas filters with small selectivity tend to miss texture details with the advantage of avoiding ringing. This results from the non-stationary nature of image statistics, and encourages us to locally select the most suitable filter type for each region in the image. In Fig. 1 (c) we show how this strategy allows to reconstruct texture in areas with small contrast and avoids ringing in regions with high contrast (e.g. around edges).

We choose the well-known *raised cosine* filter [14] to provide a range of parametric kernels with different levels of selectivity. The analytic expression of a one-dimensional raised cosine filter is

$$h_{s,\beta}(t) = \frac{\sin(\pi s t)}{\pi s t} \frac{\cos(\pi s \beta t)}{1 - 4s^2\beta^2 t^2}, \tag{1}$$

where s is the upscaling factor (the bandwidth of the filter is $1/s$) and β is the roll-off factor (which measures the excess bandwidth of the filter). Since all the upscaling and low-pass filtering operations are separable, this expression is applied for both vertical and horizontal axis consecutively. We enforce the value of β to lie in the range $[0, s-1]$, so that the excess bandwidth never exceeds the Nyquist frequency. With $\beta = 0$ we obtain the most selective filter (with a large amount of ringing) and with $\beta = s - 1$ the least selective one.

In order to adaptively select the most suitable filter from a bank of 5 filters with $\beta = \{0, \frac{s-1}{4}, \frac{s-1}{2}, 3\frac{s-1}{4}, s-1\}$, we look for the one providing minimal matching cost for each overlapping patch, as introduced below. In Fig. 2 we show the color encoded chosen filter (ranging from blue, for $\beta = 0$, to dark red, for $\beta = s - 1$) for each patch. We denote by $\mathbf{x}_{\beta,l,i}$, $\mathbf{x}_{\beta,h,i}$, $\mathbf{y}_{\beta,l,j}$ and $\mathbf{y}_{\beta,h,j}$ a low-frequency patch, the corresponding reconstructed high-frequency patch, the best matching low-resolution reference patch and the correspond-



(a) (b)

Fig. 2. Adaptive filter selection. Left, part of a super-resolved image ($2\times$ magnification). Right, selected filters from a set of 5 raised cosine filters with $\beta = \{0, 1/4, 1/2, 3/4, 1\}$. Note how the statistical distribution of the filter selection is related to the non-stationary statistics of the image.

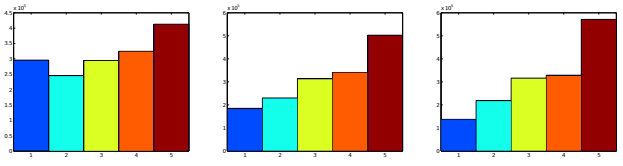
ing high-frequency example patch, respectively, which have been obtained by using the interpolation kernel and analysis filter $\mathbf{h}_{s,\beta}$. Then, we measure the local kernel cost as

$$k_{\beta,i} = \alpha\|\mathbf{x}_{\beta,l,i} - \mathbf{y}_{\beta,l,j}\|_1 + (1-\alpha)\|\mathbf{x}_{\beta,h,i} - \mathbf{y}_{\beta,h,j}\|_1. \quad (2)$$

We leave a parameter α to tune the filter selection. As shown in Fig. 3, small values of α (ignoring low-frequency differences) tend to a more uniform selection of filters, whereas large values of α (ignoring high-frequency differences) typically result in the selection of ringing-free filters, with worse separation of low and high-frequency bands. In our tests, large values of α tend to better qualitative and objective results. The final super-resolved image is obtained by averaging the overlapping patches of the images computed with the selected filters.

2.2. Implementation details

The proposed method has been implemented in MATLAB, with the costlier sections (example search, composition stages, filtering) implemented in OpenCL without special emphasis on optimization. The patch side is set to $N_p = 3$ and the search window side to $N_w = 15$. Our algorithm is applied iteratively with smaller upscaling steps ($s = s_1 s_2 \ldots$), e.g. an upscaling with $s = 2$ is implemented as an initial upscaling with $s_1 = 4/3$ and a second one with $s_2 = 3/2$.



$\alpha = 0$ $\alpha = 1/2$ $\alpha = 1$

Fig. 3. Histogram of selected filters (for $2\times$ magnification) from a set of 5 raised cosine filters with $\beta = \{0, 1/4, 1/2, 3/4, 1\}$ for different values of the tuning parameter α. The color mapping is the same of Fig. 2 (b).

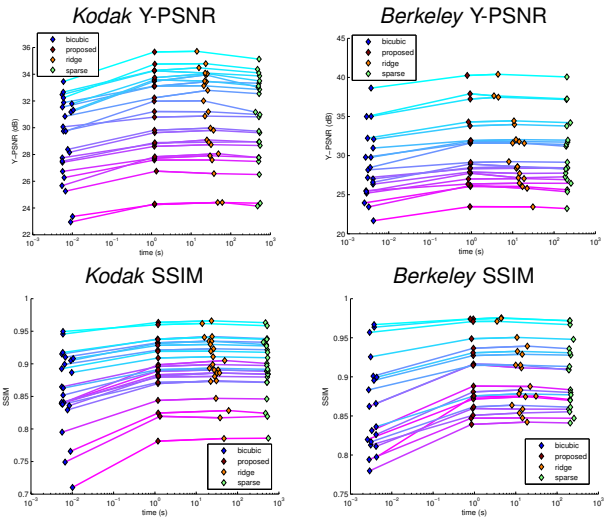


Fig. 4. Top, Y-PSNR *vs.* time for the *Kodak* (left) and *Berkeley* (right) datasets. Bottom, SSIM *vs.* time. Our proposed method is the fastest among the SR methods.

Even though the proposed method can also compute the magnification with a single step, the wider available bandwidth for matching with smaller magnification factors results in better selection of high-frequency examples, at the cost of a somewhat increased computational cost.

As a post-processing stage, we apply *Iterative Back-Projection* [1] to ensure the information of the input image is completely contained in the super-resolved one:

$$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \mathbf{h_u} * ((\mathbf{y} - (\mathbf{x}^{(n)} * \mathbf{h}_d) \downarrow s) \uparrow s). \quad (3)$$

The algorithm converges typically after 4 or 5 iterations. The upscaling ($\mathbf{h}_u$) and down-scaling ($\mathbf{h}_d$) kernels are the ones used for bicubic resizing.

3. EXPERIMENTAL RESULTS

We test our method using two different datasets. The first one, *Kodak*[2], contains 24 images of 768×512 pixels and the second one, *Berkeley*, contains 20 images of 481×321 pixels from the project website of [15] that are commonly found in SISR publications.

We compare to a baseline method (bicubic resizing) and two state-of-the-art methods falling in the subcategories of dictionary-based [8], which we refer to by *sparse*, and kernel ridge regression [11], which we refer to by *ridge*, including a powerful post-processing stage based on the natural image prior [16]. For *sparse*, we use an offline-generated dictionary obtained with the default training dataset and parameters supplied by the authors.

[2] http://r0k.us/graphics/kodak

Fig. 5. Sample results from both the *Kodak* (left) and *Berkeley* (right) datasets obtained with our proposed method. The detail pictures show a visual comparison of the groundtruth image (top left), the reconstructed one with our method (top right), *ridge* [11] (bottom left) and *sparse* [8] (bottom right). Better viewed when zoomed in.

Our comparison consists in taking each image from the two datasets, downscaling it by a factor of $1/2$ using bicubic resizing and upscaling it by a factor of $s = 2$ with each method. We measure the SSIM, Y-PSNR and execution time. The detailed results are shown in Fig. 4 and the average results for the *Kodak* and *Berkeley* datasets are shown in Tables 1 and 2, respectively. We observe all SR methods perform better than the baseline bicubic interpolation, as expected, with *ridge* and our proposed method also surpassing the dictionary-based one. This reflects the fact that dictionary-based methods do not generalize well in comparison to internal learning. In terms of execution time, our method is clearly faster than the other tested SR methods, whereas the baseline bicubic upscaling is the fastest.

In Fig. 5, we show sample results obtained from both datasets. For space reasons, we are only including 4 of the reconstructed images, but the complete test results can be found online[3]. It is worth mentioning we have not attempted to get the best possible performance by tuning any parameter, e.g. the filter selection tuning parameter (α) and the subset of roll-off factors for the available filters (β). This decision responds to our goal of making a fair, realistic comparison with the other methods, for which no parameters were adjusted.

[3]The complete resultscan be accessed from the first author's website *http://jordisalvador-technicolor.blogspot.de/2013/05/icip-2013-2.html*

4. CONCLUSIONS

We have presented a novel single-image super-resolution method suitable for interactive applications. The execution time is orders of magnitude smaller than that of the compared state-of-the-art methods, with similar Y-PSNR and SSIM scores to those of the best performing alternative [11]. Interestingly, our method's execution time is stable with respect to the reconstruction accuracy, whereas [11]'s time increases for the more demanding images. The key aspects of our proposed method are 1) an efficient cross-scale strategy for obtaining high-frequency examples based on local searches (internal learning) and 2) an adaptive selection of the most suitable upscaling and analysis filters based on matching scores. For the future work, we plan to improve the overall efficiency of the method by focusing on the filter selection stage. It would be desirable to select filters ahead of their application, which might be achieved using sparse vector machines with a properly dimensioned training set. We also plan to study the benefits of a natural image prior [16] post-processing stage.

Method	Time (s)	Y-PSNR (dB)	SSIM
bicubic	0.007	29.10	0.86
sparse	514.7	30.53	0.89
ridge	29.13	30.81	0.90
proposed	1.193	30.68	0.89

Table 1. Average results for the *Kodak* dataset

Method	Time (s)	Y-PSNR (dB)	SSIM
bicubic	0.003	28.62	0.86
sparse	208.9	30.28	0.90
ridge	13.41	30.47	0.90
proposed	0.918	30.50	0.90

Table 2. Average results for the *Berkeley* dataset

5. REFERENCES

[1] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graph. Models Image Processing*, vol. 53, no. 3, pp. 231–239, 1991.

[2] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

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

[4] Z. Lin and H.-Y. Shum, "Fundamental Limits of Reconstruction-Based Superresolution Algorithms under Local Translation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 83–97, 2004.

[5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning Low-Level Vision," *Int. J. Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

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

[7] H. Chang, D. Yeung, and Y. Xiong, "Super-Resolution through Neighbor Embedding," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004, pp. 275–282.

[8] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[9] D. Glasner, S. Bagon, and M. Irani, "Super-Resolution from a Single Image," in *Proc. IEEE Int. Conf. on Computer Vision*, 2009, pp. 349–356.

[10] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Neighbor embedding based single-image super-resolution using Semi-Nonnegative Matrix Factorization," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2012, pp. 1289–1292.

[11] K. I. Kim and Y. Kwon, "Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.

[12] J.W. Glotzbach, R.W. Schafer, and K. Illgner, "A method of color filter array interpolation with alias cancellation properties," in *Proc. IEEE Int. Conf. on Image Processing*, 2001, vol. 1, pp. 141–144.

[13] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. on Graphics*, vol. 30, pp. 12:1–12:11, 2011.

[14] Y. Lin, H. H. Chen, Z. H. Jiang, and H. F. Hsai, "Image resizing with raised cosine pulses," in *Proc. Int. Symposium on Intelligent Signal Processing and Communication Systems*, 2004, pp. 581–585.

[15] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[16] M. F. Tappen, B. C. Russell, and W. T. Freeman, "Exploiting the Sparse Derivative Prior for Super-Resolution and Image Demosaicing," in *Proc. IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.