Explore the Power of External Data in Denoising Task

Yipin Zhou*
Brown University

Abstract

The goal of this paper is to explore the power of external data in the image denoising task, that is, to show that with taking advantage of an immense amount of information provided by external datasets, external denoising method should be more promising than internal denoising method which only extracts information from the input noisy image itself. In this paper, we present a simple external denoising method which combines Non Local Means (NLM) [Buades et al. 2005] with a randomized patch matching algorithm [Barnes et al. 2009] to denoise the input image (with an large enough external dataset) efficiently. Experimental results on a large set of images demonstrate that this external denoising method can outperform the according internal NLM and be competitive with the method [Mosseri et al. 2013] which properly combine the denoising results of both internal and external NLM. However, one drawback of the external denoising method is that compared with internal method, it is more vulnerable to noise overfitting problem. At the end of the paper, we also discuss a possible extension — applying adaptive patch size during denoising to reduce the overfitting problem and to make the external denoising method even more powerful.

Keywords: external image denoising, Non Local Means, PatchMatch, internal image denoising

1 Introduction

Image denoising, the process of recovering a clean natural image from a noise corrupted one, is a long-standing and ill-posed problem.

$$I = X + n \tag{1}$$

Where *I* represents the observed image, *X* is the ground truth clean image and *n* is the noise. In most cases, we consider *n* are i.i.d. gaussian values with zero mean and known variance, which is called additive white Gaussian noise (AWG).

Image denoising algorithms have made considerable progress in the last few decades. And those algorithms can be broadly divided into two classes: Internal Denoising and External Denoising.

Internal Denoising means image patches are denoised only using the patches from the noisy image itself. NLM (non local means) [Buades et al. 2005], a simple non parametric method, denoises image patches by weighted averaging pixels from the same image. BM3D [Dabov et al. 2007] is the extension of NLM, but uses a more effective noise reduction strategy. It groups similar-looking patches from the input image into 3D data arrays and applies collaborative filtering.

External Denoising means image patch are denoised using external clean natural image patches. Those clean patches can be coming from an external database of clean images. For instance, EPLL [Zoran and Weiss 2011] denoises image patches efficiently using a patch based Gaussian mixture prior learned from a database of clean image patches. And in [Burger et al. 2012], they map noisy image patches to clean image patches. The mapping is learned with a plain multi layer perceptron (MLP) using large image databases.

There are also methods that are hard to be classified clearly. For example, KSVD [Elad and Aharon 2006] is based on sparse and redundant representations over trained dictionaries which is obtained by using K-SVD algorithm [Aharon et al. 2006]. As for the training data, it can be both from the corrupted image itself or from a external image database.

Recently, there appears work aiming to combine advantages of internal and external denoising according to some metrics to further improve the performance. [Mosseri et al. 2013] blends denoised patches from internal and external denoising methods together to form a single result based on patch-wise signal-to-noise-ratio value which they call PatchSNR. Patches with high PatchSNR (textured patches) benefit from external data, whereas patches with low PatchSNR (smooth patches) benefit more from internal data. And [Burger et al. 2013] proposes a learning based approach using a neural network, that automatically combines denoising results from an internal and an external method. Since larger patch size is applied in this work, they reach a different conclusion with [Mosseri et al. 2013]'s, where they claim external denoising is usually better on irregular and smooth regions while internal denoising prefers regular and repeating structures.

Internal and external denoising have their own advantages and disadvantages. The former only extracts information from noisy image itself, which is relatively efficient but can not guarantee a rich representation, moreover the data is corrupted. While the latter contains redundant clean data and rich representation but has higher risk in overfitting the noise, especially for the smooth patches and it is also observed by [Mosseri et al. 2013]. There has been no final conclusion on which one is more superior than the other. Although the authors in the work [Zontak and Irani 2011] claim internal image-specic statistics is better than general external statistics, they use example images with regular and repeating structures which suits the internal method better. Regarding to this issue, our observation is that the disadvantages of external denoising are able to be improved to some degree. For example, many algorithms, such as KD-Tree [Bentley 1975] and coarse-to-fine method can be applied to accelerate the searching in large external databases. Furthermore, parametric techniques usually spend most time on training procedure (learning image or patch based priors using large datasets) and denoising phase is relatively efficient. And the overfitting problem can also be reduced by some ways, for instance, applying adaptive patch size [Levin et al. 2012]. However, the lack of representation and corruption for internal data is unavoidable. So we claim that with good use of redundant data from external datasets, external

*e-mail: yipin_zhou@brown.edu

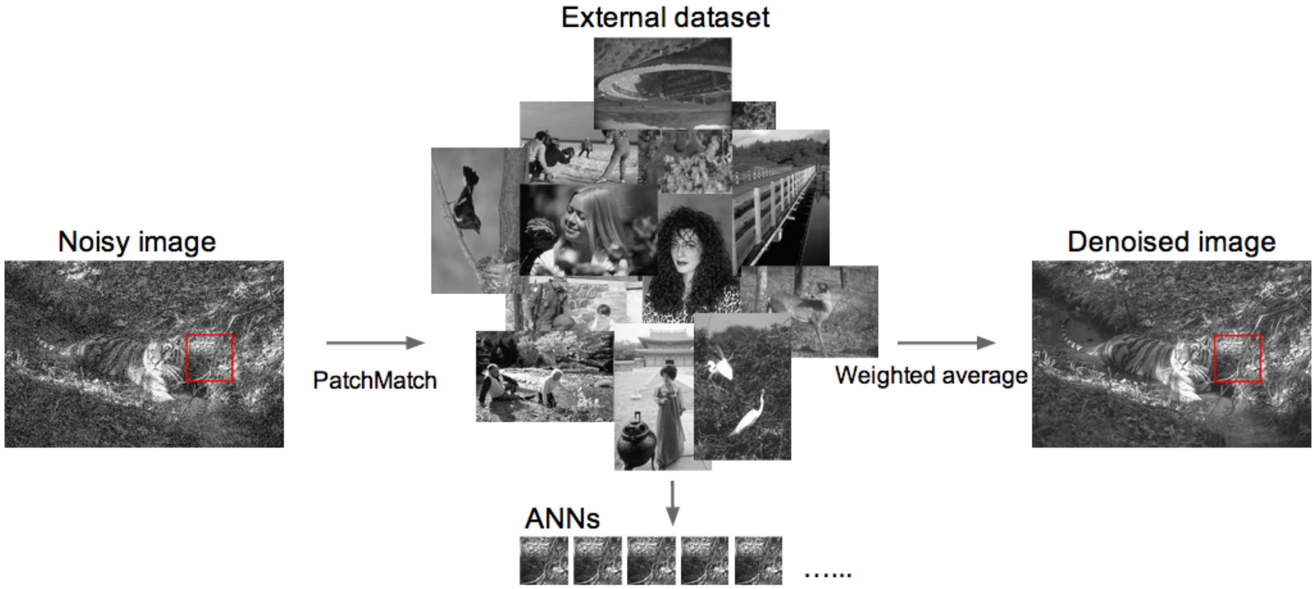


Figure 1: *Overview of our denoising method. For each patch in the noisy image (eg. the red square), we search approximate nearest neighbors from the external dataset containing clean images through PatchMatch algorithm. And those ANNs will be weighted averaged into a denoised patch. Those generated denoised patches will form the final result.*

data can be more powerful than internal data in denoising task.

To support this view, in this paper, we present a non-parametric external denoising method. We combine an external NLM (instead of searching nearest neighbors of noisy patches from corrupted image itself, we search from a large enough external database) with PatchMatch [Barnes et al. 2009] algorithm, which finds approximate nearest neighbors (ANNs) of noisy image patches. Compared with exact nearest neighbors results, results consisting of ANNs patches are smoother and survive less noise overfitting. More important, this randomized patch matching algorithm largely accelerate the denoising procedure.

We evaluate our method on a broad range of images and find that our results are obviously better than the internal NLM implemented by [Buades et al. 2005] and also competitive with the results combining the best of internal and external NLM implemented by [Mosseri et al. 2013]. However, our external denoising method still suffer from noise overfitting, especially for the smooth patches. We also explore one of possible way — using adaptive patch size to improve overfitting problem by making the so-called *preference image* experiment to show the promise of this extension.

2 Overview

An overview diagram of our approach is shown in Figure 1. Under the NLM framework, for each 7x7 patch in the noisy image, we apply PatchMatch method to search enough number of approximate nearest neighbors from the external dataset. Then we weighted average those ANNs into a denoised patch. And generated denoised patches will form the final denoised image. The rest of the paper is organized as follows: Sec. 3 analyzes the denoising algorithm by explaining the two main parts NLM and PatchMatch method and also some implementation tricks. Sec. 4 evaluates the denoised results generated by our method and prove our external denoising method can be superior compared with internal NLM and other state-of-the-art methods. Sec. 5 propose one possible extension to reduce the noise overfitting problem, especially for the smooth patches and demonstrates it by a *preference image* experiment. Finally we conclude and discuss other possible future research directions in Sec. 6.

3 Denoising algorithm

3.1 External non local means algorithm

In the original work of [Buades et al. 2005] or say internal NLM, they explore the self-similarities of the natural images. Each pixel in denoised image is obtained as a weighted average of pixels centered at regions in the noisy image which are similar to the region centered at the estimated pixel in the same noisy image.

Instead of pixelwise estimating the denoised image using only noisy image itself, in our external NLM method, we estimate each denoised patch by weighted averaging patches from the external database containing large number of natural images, which are similar with the estimated patch from noisy image. Concretely, we consider P_{noise}[i] is a patch centered on i-th pixel in the input noisy image. And $P_{external}$[i, j] represents j-th approximate nearest neighbor patch (from the external dataset) of P_{noise}[i]. While $P_{denoise}$[i] is according denoised patch of P_{noise}[i]. We have the equations as following:

$$P_{denoise}[i] = \sum_{j=1}^{n} W[i,j] P_{external}[i,j], \tag{2}$$

$$W[i,j] = \frac{1}{N[i]} e^{-\frac{K_b(P_{noise}[i] - P_{external}[i,j])}{h^2}}, \tag{3}$$

where K_b is the Gaussian kernel with bandwidth b, parameter h is the degree of filtering. And N[i] is the normalizing constant.

$$N[i] = \sum_{k=1}^{n} e^{-\frac{K_b(P_{noise}[i]-P_{external}[i,k])}{h^2}} \tag{4}$$

W[i, j] depends on the similarity between the patch P_{noise}[i] and the patch $P_{external}$[i, j]. Besides it satisfies $0 \leq$ W[i, j] ≤ 1 and also $\sum_{j=1}^{n}$ W[i, j] = 1. And the similarity is described as a decreasing function of the Gaussian weighted Euclidean distance between two patches. That is to say, the larger the distance between two patches, the smaller the weight, and vice versa. One can reference Figure.2 for an example.

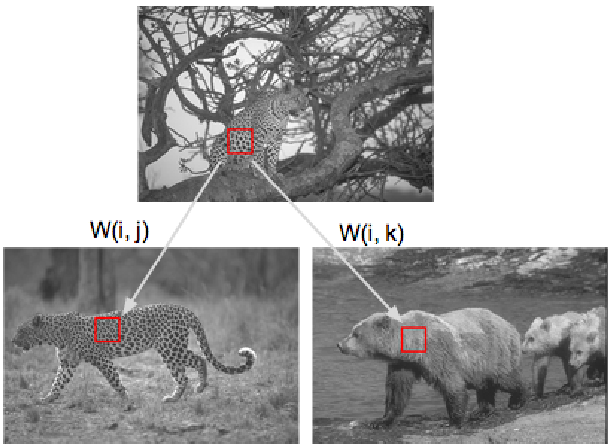


Figure 2: *Similar patch neighborhoods give a large weight W(i, j), while much different neighborhoods give a small weight W(i, k).*

This method recovers the noisy image patchwisely, for those nearby patches that have overlap regions, we average the values within those overlap regions (maintaining the consistency of the recovered image) to compute a single value for each pixel. And since the external database is not unlimited, for a noisy patch if there is no qualified nearest neighbor patch (the distance between two patches that is beyond a noise variance based threshold), we just use the noisy patch directly as a denoised patch in the result image.

3.2 A randomized patch matching algorithm

For our denoising algorithm, finding nearest neighbors (NNs) or approximate nearest neighbors (ANNs) of a noisy patch from a large external dataset in a fast speed is extremely important. If we exhaustively search NNs for each patch in the noisy image, it may take days or even weeks to process one single image. It is obviously unpractical.

In [Zontak and Irani 2011] and [Mosseri et al. 2013], they implement an external denoising algorithm using KD-Tree to search ANNs from external random patch databases. While for a large dataset, even applying KD-Tree method is not efficient enough to limit the running time into a reasonable range. In this paper, we use a randomized patch matching method which is called PatchMatch to find ANNs efficiently from external datasets. This method is proposed by [Barnes et al. 2009], and it's ability in finding qualified ANNs in fast speed enabling its application in real-time editing, such as image reshuffling, image completion and image retargeting. Since PatchMatch makes use of natural coherence in the imagery to propagate matches to surrounding areas, we collect our external dataset as group of natural images instead of random patches.

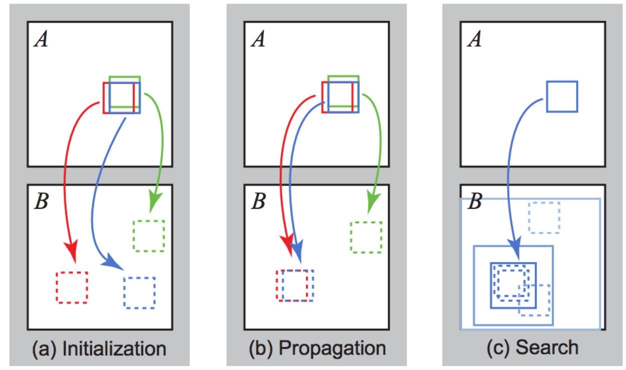


Figure 3: *Phases of the randomized nearest neighbor algorithm: (a) patches initially have random assignments; (b) the blue patch checks above/green and left/red neighbors to see if they will improve the blue mapping, propagating good matches; (c) the patch searches randomly for improvements in concentric neighborhoods.*

In this algorithm that computes patch correspondences, they define a nearest-neighbor field (NNF) as coordinates offsets. That offset means, for example, given patch coordinate (x_a, y_a) in image A and its corresponding nearest neighbor coordinate (x_b, y_b) in image B, the offset *f*(x, y) is (x_b-x_a, y_b-y_a). To compute a good qualified approximate NNF, The algorithm has three main components: Initialization, Propagation and Random search. The latter two form an iteration process. Figure. 3 is originally from [Barnes et al. 2009], it illustrates those three steps as an example. And below, we will discuss those steps in detail.

Initialization The nearest-neighbor field can be initialized by using some prior information or just by assigning random values. In this task, we initialize NNF by sample values across the full range of image B randomly.

Propagation This is an iterative update process. We propagate by examining offsets in scan order (from left to right, up to down).To improve *f*(x, y), we compute the patch distance of *f*(x, y), *f*(x-1, y) and *f*(x, y-1). And use the best one to improve *f*(x, y). For instance, *f*(x-1, y) is the best, that means the patch at (x-1, y) that is one pixel left at the estimated patch (x, y) has a relatively better mapping. Then we use the patch that is one pixel right at the mapping of patch (x-1, y) as the new mapping of the estimated patch (x, y). Moreover, on even iterations we propagate by examining offsets in reverse scan order, using *f*(x+1, y) and *f*(x, y+1) as our candidate offsets.

Random search This is also an iterative process together with the last step. In this part, we randomly search the the neighborhood of the best offset found so far to further improve the offset *f*(x, y). According to [Barnes et al. 2009], we name the sequence of candidate offsets to be tested as f_i(x, y). And we have:

$$f_i(x, y) = f(x, y) + R_i \alpha^i w, \tag{5}$$

where w is a large maximum search radius, R_i is random in [-1, 1]×[-1, 1] and α is a fixed ratio range from (0, 1). We examine patches for i = 0,1,2...until the current search radius α^iw is below 1 pixel. In this task we set α = 0.5 and set w the maximum image dimension.

For more specific analysis about PatchMatch, one can refer [Barnes et al. 2009]. In this work, we try to find ANNs by processing external images one by one and sort those candidate ANNs then pick the number we need. We also try to stick external images together to form a large one and search enough number of ANNs directly. And we find that the latter produce better result, so we apply this trick in our denoising method.

4 Evaluation

To evaluate our denoising method and prove the power of using external data, we first compare our method with the traditional (internal) NLM [Buades et al. 2005] (we call it Internal NLM) and then with the results of properly combining the internal and external NLM implemented by [Mosseri et al. 2013] (we call it Combining NLM) and their external method apply KD-Tree to search ANNs from a external database.

Regarding the implementation issues, for all methods (Internal NLM, Combining NLM and our method) the patch size is 7×7. And external database, both our method and external part of Combining NLM use 200 training images of Berkeley Segmentation Dataset (BSDS300) [Martin et al. 2001] which means it contains around 30 millions of random 7×7 patches.

We evaluate those method using 100 testing images from BSDS300 and add white Gaussian noise with the variance equaling to 35. Our method takes about 40 minutes to run one single image on my laptop, while the Combining NLM takes over night and the brute force implementation of external NLM will take several days. Figure. 4 shows part of results generated by Internal NLM, Combining NLM and our method and their PSNR values. As we can see, compared with internal denoising method, external method is better at maintaining the details of the images because of their rich representation of patches, but suffer more with noise overfitting problem, especially for smooth areas, such as Figure 4.15 and Figure 4.20.

We also compute average PSNR values of these 100 testing images for Internal NLM, combining NLM and our method. Table. 1 shows the results, we can see that our method outperforms Internal NLM and is competitive with Combining NLM which has been claimed combining the advantages of internal and external denoising method. Though most testing images contain large smooth areas, external method can still generate better (visually and quantitatively) results which proves the power of applying external data in denoising task.

Table 1: *Average PSNR values: Denoising by appling a small gaussian blur (1stcol); Internal NLM (2^{ed}col); Combining NLM (3rdcol); Our results(4thcol).*

Gaussian blur	Internal NLM	Combining NLM	Ours
24.4293	26.0281	26.6376	26.6208

5 Adaptive patch size

From Figure. 4 we can notice that noise overfitting in external denoising method is more severe than in internal method, especially for the images with large smooth areas. And this is the main disadvantage of external method. If noise overfitting problem can be alleviated to some degree, the using of the external data will become more powerful.

Since the patch size we currently apply is very small, according to [Levin et al. 2012]' s work, it is possible to alleviate overfitting problem and improve the PSNR value of the final results by increasing the patch size and smooth patch will benefit more (alleviating overfitting problem and improving PSNR value more) from this increasing. Nevertheless, larger patch size means a more complicated representation, that is, a much richer representation dataset is needed if we want to guarantee enough number of qualified NNs or ANNs for noisy patches. So only external method with large enough datasets has this qualification (the representation in internal method is always limited because the 'dataset' is the noisy itself).

However, the increase in patch size requires a much larger increase of database to guarantee enough good NNs and infinite database is impractical. Intuitively, the database will run out of detailed patches (textured patches) first, then smoother patches. And smooth patches have higher risk to suffer from noise overfitting problem and benefit more from patch size increasing, like we just said. To balance the patch size and external data size limitation problem, it is a good choice to apply an adaptive method, that is, using larger patch size for smoother patch and smaller patch size for detailed patch, which is also mentioned in [Levin et al. 2012]. By doing so, for detailed patches (have lighter overfitting problem), the smaller patch size make them easier to find enough good representations in a limited dataset. And for smooth patches (suffer more from overfitting), the larger patch size will bring more improvement and do not need a significant dataset expansion because of their simple structures. Adaptive method makes it possible to have most significant PSNR gain while keeping the external datasets a reasonable size.

To demonstrate the good prospects of this extension, we do a simple experiment as the Figure. 5 shows. We pick some images and denoise them with exactly the same external dataset again by applying a larger patch size (15×15). Then we compare large patch version and small patch version with the ground truth image pixelwisely and draw the preference image (blue pixel means the pixel value of large patch version is closer to ground truth, and red pixel means the pixel value of small patch version is closer to ground truth) as the fourth column in Figure. 5 shows. From the preference image, we notice that large patch version works better in smooth areas and small patch version generate better results in textured regions. Finally we simply combine the results generated by two patch sizes based on the pixelwise preference image, that is, keeping the pixel value which is closer to the ground truth. Fifth column shows the combining results, the noise overfitting problem is largely alleviated.

The goal of this experiment is to show the promise of applying adaptive patch size approach. To make it a concrete work, more things need to be done in the future. For instance, what is the mathematical relationship between structural complexity of a patch (measured by gradient value or other descriptors) and it's optimal size. We believe this extension is able to significantly improve the quality of denoised results and make the external method even more powerful.

6 Conclusion

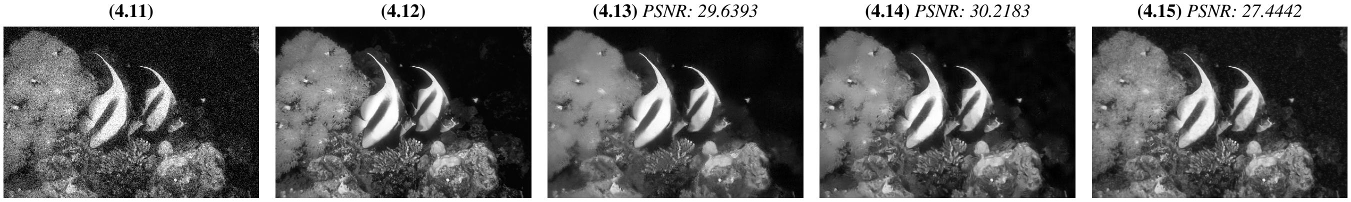
In this paper, we present a external denoising algorithm combining external NLM and PatchMatch searching algorithm. Even though this approach is quite simple, it outperforms Internal NLM easily and is competitive with the method combining Internal and external NLM properly. Those experiments demonstrate that applying external data in denoising task is more promising and powerful than internal data, though the patch noise overfitting problem is more severe for external approach. To alleviate this problem, we propose an extension — applying adaptive patch size. And we prove the good prospects of this direction and discuss some possible future work.

The goal of this paper is to prove the power of using external data

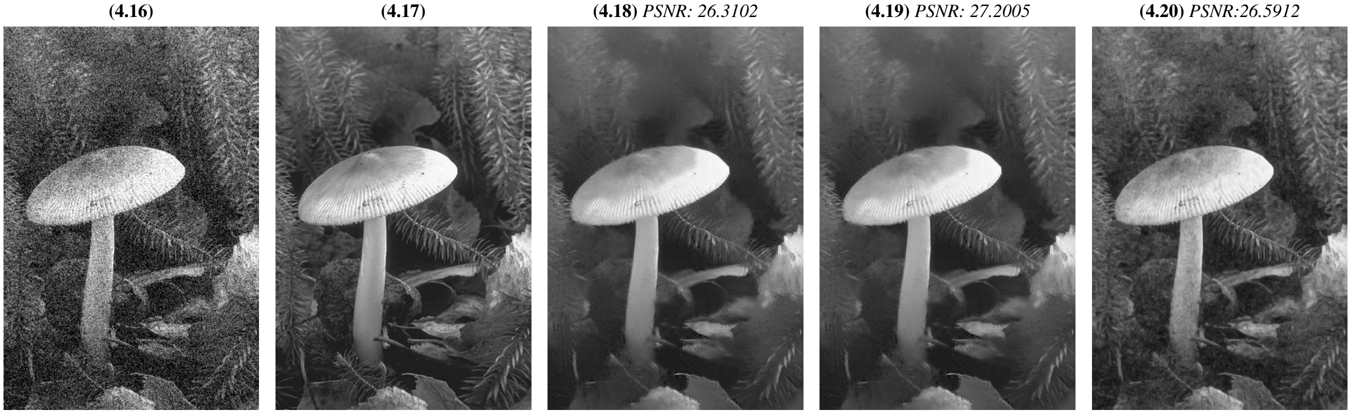
(4.1) **(4.2)** **(4.3)** *PSNR: 23.8660* **(4.4)** *PSNR: 24.6783* **(4.5)** *PSNR: 27.4453*

(4.6) **(4.7)** **(4.8)** *PSNR: 25.4447* **(4.9)** *PSNR: 26.2921* **(4.10)** *PSNR: 27.7214*

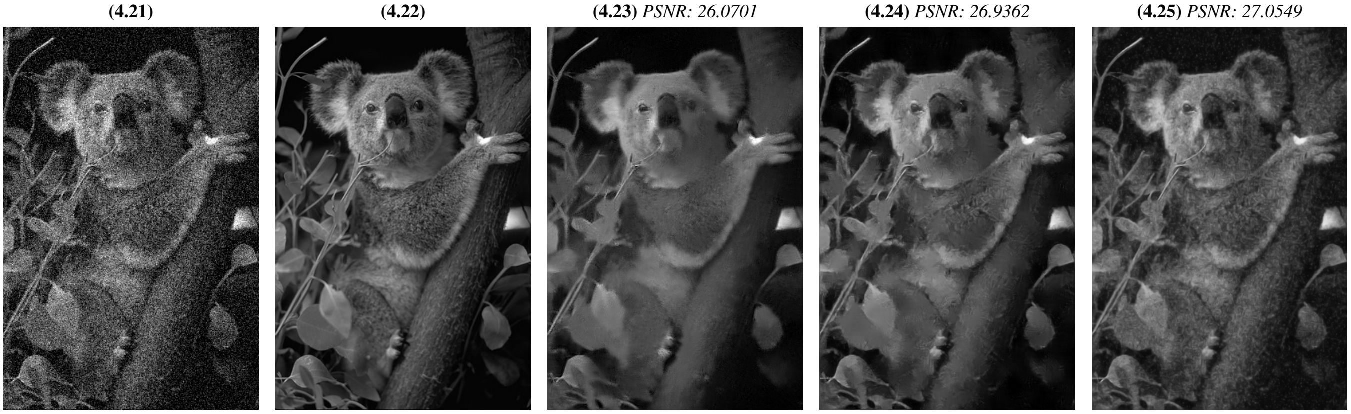
(4.11) **(4.12)** **(4.13)** *PSNR: 29.6393* **(4.14)** *PSNR: 30.2183* **(4.15)** *PSNR: 27.4442*



(4.16) **(4.17)** **(4.18)** *PSNR: 26.3102* **(4.19)** *PSNR: 27.2005* **(4.20)** *PSNR:26.5912*



(4.21) **(4.22)** **(4.23)** *PSNR: 26.0701* **(4.24)** *PSNR: 26.9362* **(4.25)** *PSNR: 27.0549*



(4.26) **(4.27)** **(4.28)** *PSNR: 26.9766* **(4.29)** *PSNR: 27.7412* **(4.30)** *PSNR: 26.5559*

Figure 4: *Examples of denoising results. 1st column: Input noisy images; 2ed column: Ground truth images*
3rd column: [Buades et al. 2005] results (Internal NLM); 4th column: [Mosseri et al. 2013] results (Combining NLM); 5th column: Our results.

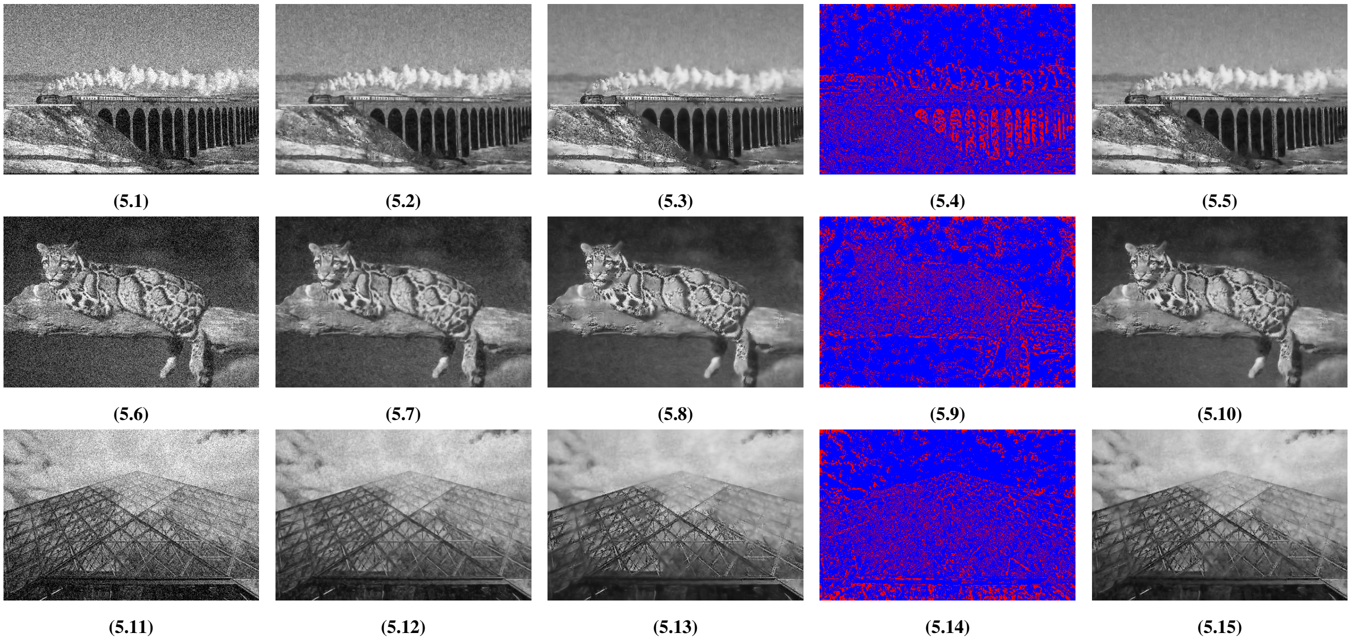


Figure 5: *Columns of results: Noisy images; Small patch version; Large patch version; Preference images; Combining images.*

in denoising task. We apply a non-parametric approach and have to traverse the whole database every time when processing the noisy image. With largely increasing of the database size, the limitation of the memory and the running speed make this algorithm not practical enough anymore, which limits the power of richly external data. Under this situation, parametric approaches can probably complementary, though finding optimal image or patch based prior is still an open problem [Levin and Nadler 2011]. We believe that utilizing the power of large external data and accurate image priors have strong potential to further improve the denoising results and also other low-level vision tasks, and it needs more work to be done in the future.

References

AHARON, M., ELAD, M., AND BRUCKSTEIN, A. 2006. k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on 54*, 11 (Nov), 4311–4322.

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (Aug.).

BENTLEY, J. L. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM 18*, 9 (Sept.), 509–517.

BUADES, A., COLL, B., AND MOREL, J. M. 2005. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 60–65 vol. 2.

BURGER, H., SCHULER, C., AND HARMELING, S. 2012. Image denoising: Can plain neural networks compete with bm3d? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2392–2399.

BURGER, H., SCHULER, C., AND HARMELING, S. 2013. Learning how to combine internal and external denoising methods. In *Pattern Recognition*, vol. 8142 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 121–130.

DABOV, K., FOI, A., KATKOVNIK, V., AND EGIAZARIAN, K. 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on 16*, 8 (Aug), 2080–2095.

ELAD, M., AND AHARON, M. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on 15*, 12 (Dec), 3736–3745.

LEVIN, A., AND NADLER, B. 2011. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2833–2840.

LEVIN, A., NADLER, B., DURAND, F., AND FREEMAN, W. T. 2012. Patch complexity, finite pixel correlations and optimal denoising. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, Springer-Verlag, Berlin, Heidelberg, ECCV'12, 73–86.

MARTIN, D., FOWLKES, C., TAL, D., AND MALIK, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 416–423.

MOSSERI, I., ZONTAK, M., AND IRANI, M. 2013. Combining the power of internal and external denoising. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, 1–9.

ZONTAK, M., AND IRANI, M. 2011. Internal statistics of a single natural image. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 977–984.

ZORAN, D., AND WEISS, Y. 2011. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 479–486.