A Parallel Algorithm for Multi-view Image Denoising

Aashish Thite and Li Zhang

Department of Electrical and Computer Engineering
University of Wisconsin - Madison
{aashish | lizhang}@cs.wisc.edu

Abstract

In this paper, we propose to improve the denoising performance by exploiting the redundancy provided by the multiple views. Here, we review the rich literature on image denoising methods. Zhang et. al. in [1] conjecture that single view image denoising algorithms have reached the limit of their performance. [1] also formulates the problem of multiple view denoising and give an algorithm to achieve the same. Their results prove the improved performance attained because of the information in the additional views. In this paper, we propose a novel approach towards this problem of denoising images using multiple views. We use an adaptation of the NL-means denoising algorithm on images focused at different depths or as we call them, focal images. These focal images are constructed using the multiple views. We introduce the notion of super pixels that constitute the focal images. The NL-means denoising algorithm denoises these super-images. Depth values are simultaneously estimated. Each view is reconstructed using these denoised super-images and the depth map. This intuitively parallel algorithm is implemented on GPU. We present the details of our implementation. The results of our experiment not only validate our hypothesis of improved performance due to multiple views, but also show that our GPU implementation is faster than other algorithms which have comparable performance. We compare the performance of our algorithm with the state-of-the-art single view image denoising and multiple view denoising algorithms.

Keywords- multi-view denoising, non-local means, noisy images, patch-based denoising

Introduction

Motivation

A large number of today's applications in computer vision use multiple view imaging. 3D object reconstruction and recognition, motion analysis, multi-view laparoscopy and video surveillance are among the few popular of these applications. Low noise large-depth-of-field images are critical for performance of these applications. Large depth-of-field images can be captured using an array of cameras assembled together as a compact unit as explained in [1], instead of using specialized Light field cameras like [14]. The cameras used in these applications work with a small aperture and short exposure to reduce motion-blur and defocus. These could be miniature low power cameras such as those on cellphones. This setting causes the images from the cameras to be noisy.

The goal of our algorithm is to restore each of the images by jointly denoising image patches in different viewpoints. Multi-view denoising has advantages over single view denoising and video denoising algorithms. We group similar patches in the different views corresponding to particular depth values. We have more redundancy as compared to single view denoising so multi-view denoising performs better than its single view counterpart. Multi-view denoising uses a single depth map estimate as opposed to video denoising, where motion between cameras have more degrees of freedom.

Noise Model

$u(i)$ is a measurement of intensity of light at CCD captor in a CCD matrix of a camera. $n(i)$ is the perturbation in the observed value $v(i)$ caused by the digitization process as explained in [9]. We assume that $n(i)$ and $n(j)$ values at different pixels are independent or uncorrelated. We also assume that the image is a stationary random process.

$$v(i) = u(i) + n(i)$$

A simple way to model noise in an image is by adding independent and identically distributed zero mean Gaussian noise with variance σ^2.

We have a set of observations of the same scene captured from different viewpoints. The formulation of the problem is similar to the formulation in [1]. We have $\{I_m\}_{m=1}^{M}$ as a set of M noisy images captured from M

viewpoints at the same time instant. $\{G_m\}_{m=1}^{M}$ is a set of underlying noiseless images and n_m is a zero mean Gaussian noise.

$$I_m = G_m + n_m$$

Our goal is to estimate these underlying images given the observed noisy set of images.

Approach

We review the literature on the popular approaches to image denoising. Though our formulation of the problem is similar to Zhang et. al. [1], our assumptions and approach towards it is quite different. Zhang et. al. in [1] assume the noise in the images to have a Poisson distribution while we assume it to be white Gaussian. Using our noisy set, we first construct a set of images focused at depths corresponding to integer disparities between the multi-view images. We call these focal images. These focal images consist of super pixels or every pixel in the focal image has multiple pixels. We then denoise these focal images consisting of super pixels using an adaptation of NL-means denoising algorithm that and simultaneously estimate the disparity map. Now using the estimated disparity map, we can extract denoised multi-view images from the denoised focal images. This algorithm is easily parallelized and is implemented on GPU. Our algorithm runs faster as compared to [1] and shows comparable performance with respect to peak signal to noise ratio (PSNR).

Related Work

Image denoising problem has a whole spectrum of approaches with a rich literature available. These works originated from various fields like statistics, signal processing and multi-resolution analysis and have an assumption about the underlying signal. These work in either in the transform domain or spatial domain. In the transform domain, the signal is represented in sparse form. By suppressing the weak coefficients in this transformed signal as high magnitude components belong to the true signal. The image is transformed back from transform domain to get the estimated denoised image. The spatial domain denoising methods exploit the fact that image patches repeat in the spatial locality. These patches grouped together and are collaboratively denoised. [3] is an excellent review for the work done in the domain of image denoising. Although these algorithms use different techniques and tools, the basic idea behind image denoising is averaging. [11] introduces the locally adaptive filters. The noisy image is sampled by a moving window and the DCT transform of every sample is modified. [10] uses a similar approach but uses shape-adaptive transform to get a sparse representation of the true signal. Elad et. al., in [12], use K-SVD and learned dictionaries to denoise images. BM3D in [4], uses stacks together similar patches in the image. Due to similarity of the patches in the stack, the 3D transform of this stack is very sparse. They perform collaborative filtering to shrink the 3D transform spectrum of this stack. The inverse transform returns a patch that can be replaced at the original locations of the patches in the group. Dabov et. al. proposed an extension to this algorithm to videos in [5]. Non-Local Mean algorithm in [2], is a spatial domain approach. It computes patch wise estimates using the weighted average of patches in the neighborhood. This algorithm is explained in detail later in the paper.

These methods are not sufficient for multi-view denoising as they are based on the assumption of similar patches existing at different locations only within the image. Zhang et. al. in [1] conjectured that limit of the performance of single image denoising has been reached. The more aggressively you denoise, more the details in the image get blurred. This is defined as method noise in [3]. Extensions of single view denoising methods are proposed to denoise videos. These algorithms search for similar patches in the adjacent frames to take part in the denoising. Vaish et. al. in [6] give an algorithm to denoise using multiple views. However they only use patches corresponding to different viewpoints and neglect locally similar patches. Chan et. al. give a multi-view algorithm based on NL-Mean algorithm. They pre-denoise the single views to estimate depth maps and iteratively improve the denoised estimates. They in their results use standard deviation values for noise up to 20. Our results show the performance with standard deviation values up to 50. Zhang et. al. in [1] give another algorithm to denoise images of the same scene from different viewpoints. It gives a maximum likelihood of denoised images given the observed noisy images and a depth map. It uses an estimated depth map as a constraint to group together k most similar patches. As in BM3D, these similar patches have low dimensionality. It uses PCA and tensor analysis of these grouped patches to reduce noise. This algorithm cannot be parallelized and it requires a good depth estimate prior to grouping patches.

Our algorithm uses a similar principle as [1] which is exploiting the redundancy with the multiple views. It uses adaptation of the NL-means algorithm which is highly parallel and does not require a prior estimation of the depth map. Like [1], our algorithm will work on off the shelf cameras. Our experiments show that our algorithm performs better than single view denoising algorithms and as good as [1] with respect to PSNR.

Multi-view Image Denoising

In this section we describe in detail our proposed algorithm. Following are the broad steps in this algorithm.

- *Construction of Focal Images*: We construct a set of images focused at depths corresponding to integer disparities in the given multi-view noisy images.
- *Non-Local Mean Denoising*: These focal images are denoised using an adaptation of the NL-means denoising algorithm.
- *Estimating the Disparity Map:* Along with denoising, we simultaneously estimate the disparity map.
- *From Denoised Focal Images to Denoised Multi-views:* We now use the disparity map and the denoised focal images to merge together denoised multi-view images.

Construction of Focal Images

We have a set of M views of the same scene, represented as $\{I_m\}_{m=1}^{M}$. In this step we create a set of images $\{F_d\}_{d=0}^{15}$ where F_d is an image focused at a depth corresponding to disparity d. We work with disparities up to 15.

F_d is not a regular image but a super Image made up of super pixels. A super pixel is a pixel made up of multiple pixels. A super pixel in a focal image is made up of pixels picked up from each of the input multi-view images at locations depending upon the disparity we are working with. The reason for using super pixels is explained in the next step of the algorithm. Thus, the number of pixels in a super pixel depends upon M, the number of views we are using. The index of the pixel in a super pixel indicates which camera it came from.

Figure 1 shows a 1D illustration of constructing a focal image using 3 views. We have three cameras in the same plane and separated by equal distances from the central camera. A super pixel in F_d at location p will consist of pixels at location p from camera C, p+d from C+1 and p-d from C-1. Similarly if there are cameras C+2 and C-2, the super pixel will contain pixels at locations p+2d and p-2d from those cameras respectively. This is equivalent to drawing a line from the camera center of each of the cameras to a point at depth corresponding to disparity d and collecting pixels that the lines encountered in the camera planes. The shifted locations for pixels which lie beyond image dimensions are replaced by trivial values. Similar calculations are done for the second dimension. At the correct disparity/depth value, all the pixels in the super pixel will correspond to the same 3D point in the scene and thus will have similar color. These super pixels constitute a super image. Now, we can see that the super image or the focal image F_d is a stack of warped images.

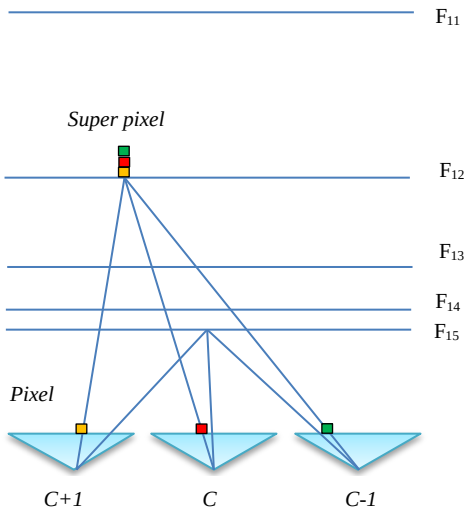


Fig1: Construction of Focal Images

Larger the disparity, closer is the focal plane of the focal image. Disparity of zero implies a focal plane at infinite depth. Figure 2 shows a visualization of a few super images when pixels in a super pixel are averaged together. We can observe that F_5 is focused on the wall, F_8 is focused on the table and the objects on it, F_{10} is focused on the face statue and F_{14} is focused on the lamp. Before using super pixel focal images in our older experiments, we worked on these averaged images. This created artifacts in the result image like color bleeding and halo effect near occlusions. This is because, if you see F_5 and F_8 in Figure 2, the orange color of the lamp bleeds outside the lamp region. These artifacts are caused by the averaging of the pixels in the super pixels with equal weight. More the difference between the true disparities across the boundary of the objects, stronger is this artifact. We will see how the super pixel approach handles this problem.

Non-Local Mean Denoising

The NL-means algorithm takes advantage of the high degree of redundancy of any natural image. A small patch in a natural image has several similar patches in in the neighborhood. Thus, NL-means is patched based spatial domain denoising approach. The neighborhood is usually a square window of predefined size around the patch under

F_5 F_8 F_{10} F_{14}

Fig2: Visualization of Focal Images:

consideration. It assumes that the image is a stationary random process.

Given a noisy observation $v(i) = u(i) + n(i)$, $NL[v(i)]$ minimizes the squared error with $u(i)$. Denoised value of a patch is estimated as a weighted average of patches in its neighborhood.

$$NL[v(i)] = \sum_{j \in neigborhood\ of\ i} w(i,j) * v(j)$$

Here $v(i)$ is a patch centered at location i and $w(i,j)$ is the weight which is a measure of similarity between patches at i and j. Higher the similarity higher should be the weight in the weighted average. It is modelled as a decreasing function of the squared difference between the two patches.

$$w(i,j) = e^{-\frac{\max(d^2 - 2\sigma^2,\ 0.0)}{h^2}}$$

Where d^2 is the squared difference between patches located at i and j. Parameter σ^2 is the variance of the calibrated noise and parameter h is the degree of filtering. It sets the degree of decay of the exponential. The weight function is set to 1 for patches with square distances less than $2\sigma^2$. This is because pixels at i and j are i.i.d random variables and the difference of these will also be i.i.d with variance $2\sigma^2$. For larger differences, the weight decrease rapidly with distance.

In our algorithm we perform such NL-mean denoising on the super images or the focal images. We have seen in the previous step that a super image is a stack of warped multi-view images. Thus, the neighborhood now consists of not only spatially neighboring patches but also the patches above and below these patches in the stacked super image. Thus we include corresponding patches from all the views in our denoising to give an improved result over the single view image denoising methods. All these patches average together to a single denoised patch at location i. Thus the denoised focal images are regular images and not super images. Since a patch covers several pixels, there are multiple denoised values for every pixel. We average these denoised values for this pixel to get the final pixel value for the focal image.

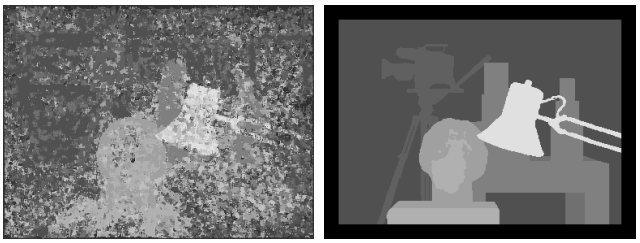
The previous section discussed the problem of color bleeding around objects in the scene that occlude other objects behind them. In case of occlusion of the part of the image in some of the views will cause the patches in the stack to be dissimilar to each other. Consequently, they will have a lower weight and thus will be treated as noise. This will reduce color bleeding due to occlusions. This however, causes ringing at the boundaries of the occlusion as less number of high weight patches take part in the denoising.

Estimating the Disparity Map

We hypothesize that for a patch location, the algorithm will find more similar patches in focal image at the correct disparity for this location than any other focal images. This for the reason we talked about in previous two sections. At correct disparity, all the patches will correspond to the same 3D point and so will be similar to each other. Consequently, the average value of the weights of the patches that take part in denoising at this location will have maximum value at the correct disparity. So we choose the disparity at which the average weight is max.

$$D(i) = \underset{d}{argmax} \left\{ \frac{1}{N} \sum_{j} w_d(i,j) \right\}$$

Here w_d is the weight calculated for a pair of patches in the focal image at disparity d. N is the total number of patches that took part in the denoising of patch at i. We could use just the total weight instead of average weight. However, in our implementation N is different for every location, how, is explained in the implementation details section later in this paper. So, we normalize this total weight with the number of patches N. Figure 3 shows the estimated disparity map and how it compares to the ground truth.



Estimated *Ground Truth*

Fig3: Disparity Map

From Denoised Focal Images to Denoised Multi-views

We now have a set of denoised focal images and an estimated disparity map. We now construct the denoised result image using these denoised focal images and the disparity map. In the previous step we saw that, at the correct disparity for a patch, we will find more similar patches than at any other disparity. Thus, the part of the image will be best denoised at this correct disparity and so we want this part of the denoised focal image to be included in the final result image. If the estimated disparity value at i is d, we pick up a patch from the denoised focal image F_d at that location. This patch will span several pixels around i depending on the patch size. Consequently, several denoised patches will overlap the pixel at location i. To determine the value of pixel in the output image, we take the average value of this pixel in the denoised patches that include this pixel. This reduces the effect of errors in the estimated disparity map. Thus, high accuracy in the disparity map estimation is not necessary to get a good result.

Implementation

In this section we provide implementation specific details of our algorithm. We use 25 views in our experiments. The focal images are constructed using these 25 views. The super pixels that constitute these focal images, thus, contain 25 pixels. The focal image structured as a stack of 25 warped images as explained in the previous section. To get a patch at location i from a particular view, we just need to use the index of this view to get the warped image and pick up a patch at location i in this image.

The denoising step in the algorithm is the most computationally intensive. For a particular location, only a few disparities have a large number of similar patches. So we reduce the number of candidate locations in the neighborhood. We first compute the weight for patch at i and the candidate patch j in the neighborhood in the image created by averaging the pixels in a super pixel, like the ones in Figure 2. Now if this weight is greater than a threshold, then to include this candidate location in the denoising step. This reduces the search space and speeds up the algorithm. This is why the number of patches taking part in the denoising is different at different locations and this is why we use average weight instead of total weight when estimating the disparity map.

By default, we use 3x3 pixel patch size, a 25x25 pixel neighborhood size, and parameter $h = 0.35$. We use Gaussian additive noise with standard deviation up to 50.

Results

In this section we present results of our experiments.

Datasets

We tested our implementation on five datasets. We added synthetic noise to these sets. Figure 4 shows a representative of each dataset. Ohta image dataset is a 5x5 viewpoint dataset and is taken from Middlebury stereo website [15]. Tarot, Euca, and Lego datasets are 17x17 image sets from Stanford Light Field Archive [16]. We only use the central 15x15 subset of these. The Text_ptgrey dataset is a 25x1 image dataset same as the one used in [1]. It was captured using PointGrey Dragon Fly Express camera at University of Wisconsin, Computer Graphics lab. The scene is about 1.5 meters from the camera and the camera moved 5mm between neighboring images.

Peak Signal to Noise Ratio (PSNR)

The PSNR gives the ratio of the maximum power of the signal to the power of noise. It is widely used as a measure of quality of denoising. It is defined using mean squared error (MSE). I is the original image, K is the image we intend to compare with the original image and N is the total number of pixels in either of the images.

$$MSE = \frac{1}{N}\sum_N [I(i) - K(i)]^2$$

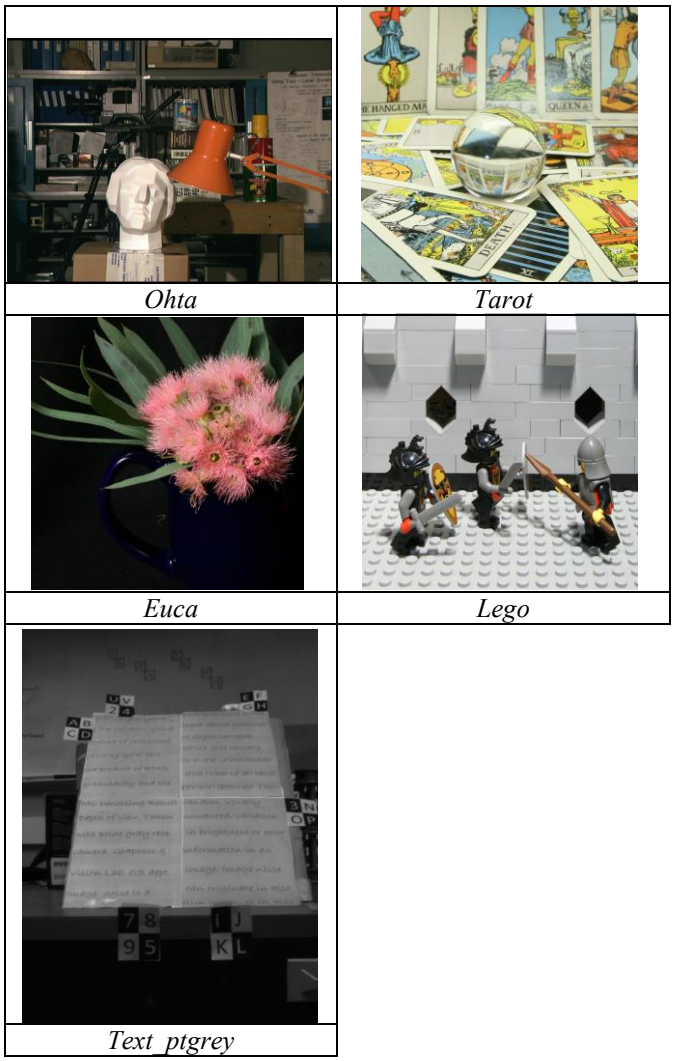


Fig4: *Datasets*

The PSNR in dB is defined as:

$$PSNR = 10 \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

MAX_I is the maximum possible intensity value in the image. It is 255 for a 24 bit color image.

Comparison with Single-view Denoising

We compare the performance of our algorithm with the state of the art single view image denoising algorithm BM3D. We use PSNR as the measure of performance. We can compare the PSNR in the Table 1 and see that our method performs better than BM3D. This performance is due to the higher number of patches taking part in the denoising in multi-views than in a single view. In Figure 5, we show a visual comparison of the results.

	Noisy	BM3D	Our Method
Ohta	18.675	27.334	30.892
Tarot	22.68	29.641	34.172
Euca	25.546	30.663	33.246
Lego	22.015	30.698	34.477
Text_ptgrey	28.715	38.264	40.153

Table1: PSNR (dB) value comparison with BM3D

Comparison with other Multi-view Denoising Method (Zhang et. al)

We compare the performance of our algorithm with the performance of [1] on the ohta dataset. [1] models noise as a Poisson distribution and we model it as a Gaussian. Table2 shows that for both the algorithms, PSNR of the output image are very close. Our implementation runs twice as fast as [1].

	Noisy	Denoised
Zhang et. al. [1]	14.147	30.659
Our Method	18.675	30.892

Table2: PSNR (dB) value comparison with Zhang et.al.[1]

Discussion and Future Work

In this paper, we use the formulation of the denoising problem in [1] and provide an alternate approach to solve the problem which has comparable performance with respect to PSNR. Our algorithm is intuitively parallel and is implemented on GPU for a faster running time. As a limitation of this algorithm, the algorithm denoises all the focal images regardless of whether or not an object is present at that disparity/depth. Also, this algorithm does not denoise parts of image which have non integral disparities. Introducing non integral disparities to our algorithm will improve the performance. Our algorithm does not model the patch warping across the views. This can be modelled as an affine transform to improve patch matching and thus improve the depth estimate.

References

[1] L. Zhang, S. Vaddadi, H. Jin, and S. Nayar. "Multiple view image denoising", CVPR, 2009.

[2] A. Buades, B. Coll; J. M. Morel, "A non-local algorithm for image denoising", CVPR, 2005.
[3] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. Simulation, 4, 2005.
[4] K. Dabov, R. Foi, V. Katkovnik, K. Egiazarian, and S. Member. “Image denoising by sparse 3d transform-domain collaborative filtering”, TIP, 16:2007, 2007.
[5] K. Dabov, A. Foi, and K. Egiazarian, “Video denoising by sparse 3d transform-domain collaborative filtering,” in Proc. European Signal Processing Conference (EUSIPCO), Sep. 2007, pp. 145–149.
[6] V. Vaish, M. Levoy, R. Szeliski, C. L. Zitnick, and S. B. “Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures”, CVPR, pages 2331–2338, 2006.
[7] M. Okutomi and T. Kanade. A multiple-baseline stereo. TPAMI, 15(4):353–363, 1993.
[8] Y. Heo, K. Lee, and S. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In CVPR, pages 1–8, 2007.
[9] G. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. TPAMI, 16(3):267–276, 1994.
[10] A. Foi, V. Katkovnik, and K. Egiazarian, Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” IEEE Trans. Image Process., vol. 16, no. 5, May 2007.
[11] L. Yaroslavsky, Local Adaptive Image Restoration and Enhancement with the use of DFT and DCT in a running window, in Proceedings, Wavelet applications in signal and image processing IV, vol. 2825 of SPIE Proc. Series, 1996, pp. 1-13.
[12] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. on Image Process., vol. 15, no. 12, pp. 3736–3745, December 2006.
[13] E. Luo, Chan, S. H., Pan, S., and Nguyen, T. Q., “Adaptive Non-local Means for Multiview Image Denoising - Searching for the Right Patches via a Statistical Approach”, Proceedings of IEEE International Conference on Image Processing (ICIP '13). 2013.
[14] R. Ng. Fourier slice photography. ACM Trans. Graph., 24(3), 2005.
[15] http://vision.middlebury.edu/stereo/data/
[16] http://lightfield.stanford.edu/

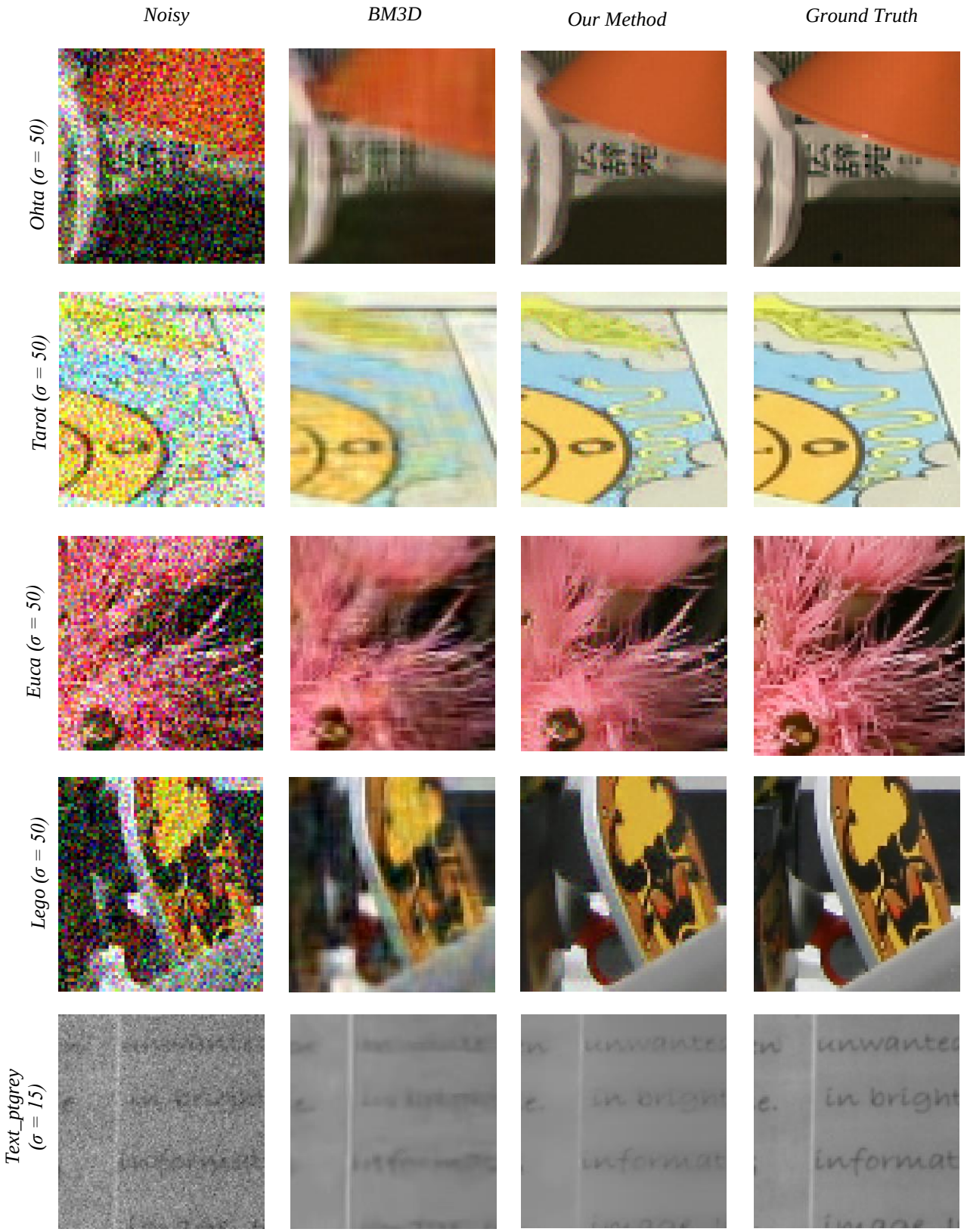


Fig5: Comparison between 25-view denoising algorithm with single view denoising algorithm BM3D