Image Denoising by Adaptive Kernel Regression

Hiroyuki Takeda, Sina Farsiu and Peyman Milanfar
Department of Electrical Engineering, University of California at Santa Cruz
{htakeda,farsiu,milanfar}@soe.ucsc.edu

***Abstract*— This paper introduces an extremely robust adaptive denoising filter in the spatial domain. The filter is based on non-parametric statistical estimation methods, and in particular generalizes an adaptive method proposed earlier by Fukunaga [1]. To denoise a pixel, the proposed filter computes a locally adaptive set of weights and window sizes, which can be proven to be optimal in the context of non-parametric estimation using kernels. While we do not report analytical results on the statistical efficiency of the proposed method in this paper, we will discuss its derivation, and experimentally demonstrate its effectiveness against competing techniques at low SNR and on real noisy data.**

I. Introduction

Classical parametric denoising methods rely on a specific model of the signal of interest, and seek to compute the parameters of this model in the presence of noise. A generative model based upon the estimated parameters is then produced as the best estimate of the underlying signal. In contrast, non-parametric methods rely on the data itself to dictate the structure of the model, in which case the implicit model is referred to as *a regression function* [2]. In particular, consider the estimation problem in two dimensions where the measured data is give by

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, 2, \cdots, P, \quad \mathbf{x}_i = [x_{1i}, x_{2i}]^T, \quad (1)$$

where y_i's are measurements, $z(\cdot)$ is the (hitherto unspecified) regression function (i.e. an unknown image) to be estimated, and ε_i's are independent and identically distributed zero mean noise values (with otherwise no particular statistical distribution assumed).

As the specific form of $z(\cdot)$ is unspecified, in order to estimate the value of the function at any point $\mathbf{x}$ given by the data, one can trust in a generic, local expansion of the function about this point. Specifically, if $\mathbf{x}$ is near the sample at $\mathbf{x}_i$, we have the N-term Taylor series[1]

$$\begin{aligned} z(\mathbf{x}_i) &\approx z(\mathbf{x}) + \nabla^T z(\mathbf{x})\,(\mathbf{x}_i - \mathbf{x}) \\ &\quad + \frac{1}{2!}(\mathbf{x}_i - \mathbf{x})^T \mathcal{H} z(\mathbf{x})\,(\mathbf{x}_i - \mathbf{x}) + \cdots \quad (2) \\ &= \beta_0 + \boldsymbol{\beta}_1^T (\mathbf{x}_i - \mathbf{x}) \\ &\quad + \boldsymbol{\beta}_2^T \mathrm{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} + \cdots, \quad (3) \end{aligned}$$

where ∇ and $\mathcal{H}$ are the gradient and Hessian operators respectively and $\mathrm{vech}(\cdot)$ is the *half-vectorization operator* [3], which lexicographically orders the "lower-triangular" portion of a matrix into a column vector.

The above suggests that if we now think of the Taylor series as a local representation of the regression function, estimating the parameter β_0 can yield the desired (local) estimate of the regression function based on the data. Indeed, the coefficients $\{\boldsymbol{\beta}_n\}_{n=1}^N$ will provide localized information on the *derivatives* of the regression function. Naturally, since the approach is based on local approximations, a reasonable step one might take now is to estimate the coefficients $\{\boldsymbol{\beta}_n\}_{n=0}^N$ from the data, giving the nearby samples higher weight than samples farther away. A least-squares formulation capturing this idea is to solve the following optimization problem:

$$\begin{aligned} \min_{\{\boldsymbol{\beta}_n\}_{n=0}^N} \sum_{i=1}^{P} \Big[& y_i - \beta_0 - \boldsymbol{\beta}_1^T (\mathbf{x}_i - \mathbf{x}) \\ & - \boldsymbol{\beta}_2^T \mathrm{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} - \cdots \Big]^2 K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x}), \quad (4) \end{aligned}$$

where $K_{\mathbf{H}}(\cdot)$ is defined as

$$K_{\mathbf{H}}(\mathbf{t}) = \frac{1}{\det(\mathbf{H})} K\left(\mathbf{H}^{-1}\mathbf{t}\right), \quad (5)$$

and called the *kernel function* [2] which penalizes distance away from the local position where the approximation is centered, and where $\mathbf{H}$ is a 2×2 "smoothing" matrix which controls the strength of this penalty. The standard choice of the matrix is $\mathbf{H} = h\mathbf{I}_2$, where h is the "global smoothing parameter". In particular, the function $K(\cdot)$ is a symmetric function, which attains its maximum at zero, and which decays away from zero at a rate controlled by the smoothing matrix. More specifically, the standard definition of the kernel function for two dimensional data has

$$\int_{R^2} \mathbf{t} K(\mathbf{t}) d\mathbf{t} = 0, \quad \int_{R^2} \mathbf{t}\mathbf{t}^T K(\mathbf{t}) d\mathbf{t} = c\mathbf{I}_2. \quad (6)$$

The choice of the particular form of the function $K(\cdot)$ is open, and may be selected as a Gaussian, exponential, or other valid forms which comply with the above constraints.

Using the matrix form, the optimization problem (4) can be posed as weighted least-squares:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \left\|\mathbf{y} - \mathbf{X}_{\mathbf{x}}\mathbf{b}\right\|_{\mathbf{W}_{\mathbf{x}}}^2, \quad (7)$$

where

$$\mathbf{y} = [y_1, y_2, \cdots, y_P]^T, \quad \mathbf{b} = \left[\beta_0, \boldsymbol{\beta}_1^T, \cdots, \boldsymbol{\beta}_N^T\right]^T, \quad (8)$$

$$\mathbf{W}_{\mathbf{x}} = \mathrm{diag}\left[K_{\mathbf{H}}(\mathbf{x}_1 - \mathbf{x}), K_{\mathbf{H}}(\mathbf{x}_2 - \mathbf{x}), \cdots, K_{\mathbf{H}}(\mathbf{x}_P - \mathbf{x})\right], \quad (9)$$

This work was supported in part by the US Air Force Grant F49620-03-1-0387, and by the National Science Foundation Science and Technology Center for Adaptive Optics, managed by the University of California at Santa Cruz under Cooperative Agreement No. AST-9876783.

[1]Other expansions are also possible, e.g. orthogonal series.

$$\mathbf{X}_{\mathbf{x}} = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x})^T & \text{vech}^T\left\{(\mathbf{x}_1 - \mathbf{x})(\mathbf{x}_1 - \mathbf{x})^T\right\} & \cdots \\ 1 & (\mathbf{x}_2 - \mathbf{x})^T & \text{vech}^T\left\{(\mathbf{x}_2 - \mathbf{x})(\mathbf{x}_2 - \mathbf{x})^T\right\} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (\mathbf{x}_P - \mathbf{x})^T & \text{vech}^T\left\{(\mathbf{x}_P - \mathbf{x})(\mathbf{x}_P - \mathbf{x})^T\right\} & \cdots \end{bmatrix}, \tag{10}$$

with "diag" defining the diagonal elements of a diagonal matrix. Regardless of the order N, our primary interest is to compute an estimate of the image (pixel values), and the necessary computations are limited to the ones that estimate the parameter β_0. Therefore, the solution for the optimization problem is simplified to

$$\hat{z}(\mathbf{x}) = \hat{\beta}_0 = \mathbf{e}_1^T \left(\mathbf{X}_{\mathbf{x}}^T \mathbf{W}_{\mathbf{x}} \mathbf{X}_{\mathbf{x}}\right)^{-1} \mathbf{X}_{\mathbf{x}}^T \mathbf{W}_{\mathbf{x}} \mathbf{Y}, \tag{11}$$

where $\mathbf{e}_1$ is a column vector with the first element equal to one, and the rest equal to zero.

Three important points are worth making here. First, the above structure allows for tailoring the estimation problem to the *local* characteristics of the data, whereas the standard parametric model is generally intended as a global fit. Second, in the estimation of the local structure, higher weight is given to the nearby data as compared to samples that are farther away from the center of the analysis window. Again this is in contrast to the general (non-adaptive) parametric approach which does not take the location of the data samples into account directly. Third, and no less important, the proposed approach is useful for both *denoising*, and equally viable for *interpolation* of sampled data at points where no actual samples exist. As such, the proposed approach is ideally suited for a wide class of image processing problems of practical interest [4].

Returning to the estimation problem based upon (4), one can choose the order N to effect an increasingly more complex local approximation of the signal. In the statistics literature, locally constant, linear and quadratic approximations (corresponding to $N = 0, 1, 2$ respectively) have been considered most widely. In particular, choosing $N = 0$ (corresponding to local constant estimation), a locally adaptive *linear* filter is obtained, which is known as the *Nadaraya-Watson Estimator* (NWE) [5]. Specifically, the estimator (11) becomes

$$\hat{z}(\mathbf{x}) = \frac{\sum_{i=1}^{P} K_{\mathbf{H}}\left(\mathbf{x}_i - \mathbf{x}\right) y_i}{\sum_{i=1}^{P} K_{\mathbf{H}}\left(\mathbf{x}_i - \mathbf{x}\right)}. \tag{12}$$

Of course, higher order approximations ($N > 0$) are also possible. What we concentrate on in the rest of this paper is the modification of the kernels. More specifically, we propose novel ways to adapt the kernels to local data. The result is a locally adaptive image filter which is able to perform denoising with high quality, even at very low SNR.

II. Spatially Adaptive Kernel Regression

A strong denoising effect can be realized by making the global smoothing parameter h larger. However, with a larger h, the estimated image will be more blurred so that we have sacrificed details to the effect of denoising. In order to have both a strong denoising effect and a sharper image, one can consider an alternative approach that will adapt the local effect of the filter using not only the position of the nearby samples, but also their gray values. That is to say, the proposed kernels will take into account two factors: spatial distances and radiometric (gray value) distances. With this idea, the kernel function can be denoted as

$$\mathsf{K}(\mathbf{x}_i - \mathbf{x}, y_i - y). \tag{13}$$

We name this the *adaptive kernel* function, and discuss the selections of the function in this section.

A. Bilateral Kernel

A simple and intuitive choice of the adaptive kernel K is to use "separable" kernels for penalizing the spatial and radiometric distances. Indeed this is precisely the thinking behind the *bilateral* filter, introduced in [6], and carefully analyzed in [7]. One of our choices is then

$$\mathsf{K}(\mathbf{x}_i - \mathbf{x}_j, y_i - y_j) \equiv K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x}_j) K_{h_r}(y_i - y_j), \tag{14}$$

where h_r is the radiometric smoothing parameter, a scalar value, that controls the rate of decay, and $K_{\mathbf{H}}(\cdot)$ and $K_{h_r}(\cdot)$ are the spatial and radiometric kernel functions, respectively. With this kernel, for the special case $N = 0$, the estimator (11) can be summarized as

$$\hat{z}(\mathbf{x}_j) = \frac{\sum_{i=1}^{P} K_{\mathbf{H}}\left(\mathbf{x}_i - \mathbf{x}_j\right) K_{h_r}\left(y_i - y_j\right) y_i}{\sum_{i=1}^{P} K_{\mathbf{H}}\left(\mathbf{x}_i - \mathbf{x}_j\right) K_{h_r}\left(y_i - y_j\right)}. \tag{15}$$

In general, the values of h and h_r are fixed. As a matter of fact, breaking K into the spatial and radiometric kernels as utilized in the bilateral case can weaken the estimator performance if SNR is very low. A simple justification for this claim comes from studying very noisy data sets, where radiometric distance $(y_i - y_j)$'s tend to be large and therefore all radiometric weights are very close to zero, and effectively useless. In the following, we present a better selection of kernels, which overcomes this difficulty.

B. Steering Kernel

The filtering procedure we propose next takes the above ideas one step further, based upon the earlier non-parametric framework. In particular, we observe that the effect of computing $K_{h_r}(y_i - y_j)$ in (14) is to implicitly measure a function of the local gradient estimated between neighboring values, and to use this estimate to weight the respective measurements. As an example, if a pixel is located near an edge, then pixels on the same side of the edge will have much stronger influence in the filtering. With this intuition in mind, we propose a two-step approach where first an initial estimate of the image gradients is made using some kind of gradient estimator (say standard kernel regression with order $N = 2$). Next this estimate is used to measure the dominant orientation of the local gradients in the image (e.g. [8]). In a second filtering stage, this orientation information is then used to adaptively "steer" the local kernel, resulting in elongated, elliptical contours spread along the directions of the local edge structure. With these locally adapted kernels, the denoising is effected most strongly

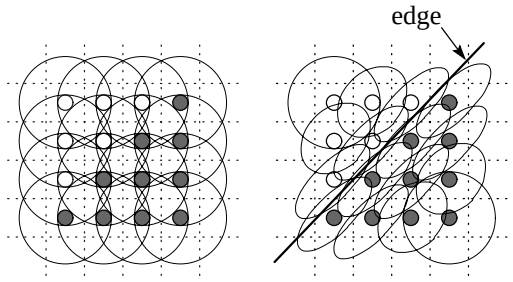


Fig. 1. Standard kernels (left) and steering kernels along a local edge (right).

along the edges, rather than across them, resulting in strong preservation of details in the final output. To be more specific, the adaptive kernel takes the form

$$\mathsf{K}(\mathbf{x}_i - \mathbf{x}, y_i - y) \equiv K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}), \tag{16}$$

where $\mathbf{H}_i$'s are the data-dependent full matrices which we call *steering* matrices. They are defined as

$$\mathbf{H}_i = h\mathbf{C}_i^{-\frac{1}{2}}, \tag{17}$$

where $\mathbf{C}_i$'s are (symmetric) covariance matrices based on the local gray-values. A good choice for $\mathbf{C}_i$ will effectively spread the kernel function along the local edges as shown in Fig. 1. It is worth noting that even if we choose a large h in order to have a strong denoising effect, the undesirable blurring effect which would otherwise have resulted, is tempered around edges with appropriate choice of $\mathbf{C}_i$'s. With such steering matrices, for example, if we choose a Gaussian kernel, the steering kernel is mathematically represented as

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) = \frac{\sqrt{\det(\mathbf{C}_i)}}{2\pi h^2} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})}{2h^2}\right\}. \tag{18}$$

The local edge structure is related to the gradient covariance (or equivalently, the locally dominant orientation), where a naive estimate of this covariance matrix may be obtained as follows:

$$\hat{\mathbf{C}}_i \approx \begin{bmatrix} \sum_{\mathbf{x}_j \in w_i} z_{x_1}^{(j)} z_{x_1}^{(j)} & \sum_{\mathbf{x}_j \in w_i} z_{x_1}^{(j)} z_{x_2}^{(j)} \\ \sum_{\mathbf{x}_j \in w_i} z_{x_1}^{(j)} z_{x_2}^{(j)} & \sum_{\mathbf{x}_j \in w_i} z_{x_2}^{(j)} z_{x_2}^{(j)} \end{bmatrix}, \tag{19}$$

where $z_{x_1}(\cdot)$ and $z_{x_2}(\cdot)$ are the first derivatives along x_1 and x_2 directions and w_i is an analysis window around the position of interest. The dominant local orientation of the gradients is then related to the eigenvectors of this estimated matrix. While this approach (which is essentially a local principal components method) is simple and has nice tolerance to noise, the resulting estimate of the covariance may in general be rank deficient, and therefore care must be taken not to take the inverse of the estimate directly in this case. In the rank deficient (or nearly so) case, a rank-one approximation will take the place of the direct inverse, or alternatively, diagonal loading or regularization methods can be used to obtain stable estimates of the covariance. In [8], we proposed an effective *multiscale* technique for estimating local orientations, which fits the requirements of this problem nicely.

In order to have a more convenient form of the covariance matrix, we decompose it into three components as follows:

$$\mathbf{C}_i = \gamma_i \mathbf{U}_{\theta_i} \mathbf{\Lambda}_i \mathbf{U}_{\theta_i}^T, \tag{20}$$

$$\mathbf{U}_{\theta_i} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}, \quad \mathbf{\Lambda}_i = \begin{bmatrix} \sigma_i & 0 \\ 0 & \sigma_i^{-1} \end{bmatrix}. \tag{21}$$

where $\mathbf{U}_{\theta_i}$ is the rotation matrix and $\mathbf{\Lambda}_i$ is the elongation matrix. Now the covariance matrix is given by the three parameters γ_i, θ_i and σ_i, which are the scaling, rotation, and elongation parameters, respectively. Fig. 2 explains schematically how these parameters affect the spreading of kernels. First, the circular kernel is elongated by the elongation matrix $\mathbf{\Lambda}_i$ and its semi-minor and major axes are given by σ_i. Second, the elongated kernel is rotated by the matrix $\mathbf{U}_{\theta_i}$. Finally, the kernel is scaled by the scaling parameter γ_i as follows. In order to reduce noise effects while producing sharp edges, wider footprint kernels are preferred in the flat areas, and smaller footprints are best in the textured areas. A simple choice of γ_i is a geometric mean of the eigenvalues of $\mathbf{C}_i$. Such γ_i makes the steering kernel area large in low frequency areas and small in high frequency areas.

III. Summary and Experimental Results

To summarize the proposed approach, we present Fig. 3(a) which displays a block-diagram of the proposed two-step approach. The first step involves computing an initial "pilot" estimate of the image gradients using a (preferably low-complexity) denoising filter and a gradient filter. For instance, the standard kernel regression with order ($N > 0$) is a simple choice for the initial gradient estimate. Other choices, such as the bilateral filter plus some kind of gradient filter are also suitable for this first stage. The next stage involves estimating the smoothing (steering) matrices from this pilot estimate. Using the local orientation information, we compute the final image by applying the steering kernel regression to the original noisy image data. At relatively high SNR, this approach will remove noise fairly well. However, at low SNR, a further refinement is called for. In this case, we can apply the same procedure iteratively as shown in Fig. 3(b). With this iterative approach, we can obtain better estimates of the image gradients. Using the gradients, the new estimated smoothing matrices are also better. Subsequently, the regression removes noise more effectively using those matrices. While we do not provide an analytical proof here, we observe that a modest number (typically less than 10) of iterations will yield a minimum mean-squared error estimate. Iterations beyond this point tend to worsen the bias in the estimate, while keeping the variance still low. The net effect of "over-iteration" being an increasingly more blurry result.

The first denoising experiment is shown in Fig. 4. We add white Gaussian noise with standard deviation of 25 (the corresponding SNR is 5.64[dB]) to the Lena image of Fig. 4(a), which gives us the noisy image shown in Fig. 4(b).

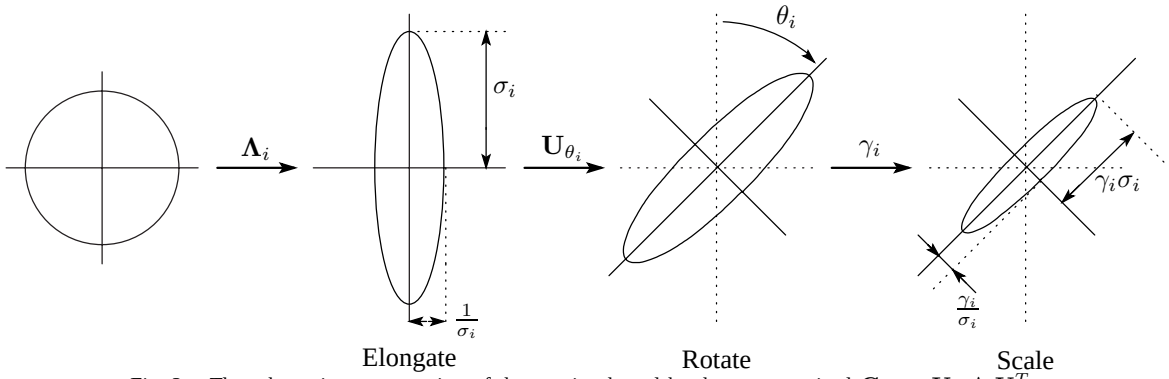
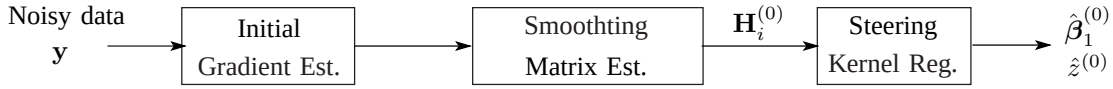
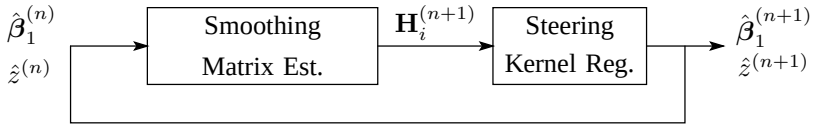


Fig. 2. The schematic representation of the steering kernel by the parameterized $\mathbf{C}_i = \gamma_i \mathbf{U}_{\theta_i} \Lambda_i \mathbf{U}_{\theta_i}^T$.



(a) Initialization



(b) Iteration

Fig. 3. Block diagram representation of the iterative denoising filter.

Fig. 4(c)[2] is the denoising result by BLS-GSM (Bayes Least Squares-Gaussian Scale Mixture) proposed by Portilla et al [9], which removes noise in the wavelet domain and is regarded as the state of art image denoising method. Fig. 4(d) is the result provided by the proposed second order steering kernel regression with $h = 2.5$ and 12 iterations. Corresponding root-mean-squared errors (RMSE) for these are (c)7.01 and (d)6.83. Our result produces very clean edges, as seen in Fig. 4(e) and (f).

In the second experiment, we remove film grain from a real image shown in Fig. 5(a). The denoising result by BLS-GSM [9], bilateral filter (15) with $h = 2.0$ and $h_r = 3.5$, and the second order steering kernel regression with $h = 2.0$ and 3 iterations are in Fig. 5(b), (c) and (d) respectively. In this case, the performance differences are easily seen in the residuals (differences between noisy data and denoised data). Fig. 5(e), (f) and (g) show the absolute values of the residuals on the luminance channel.

In conclusion, we presented a novel ("universal") non-parametric denoising algorithm, and experiments on simulated and real data attest to the superior performance of this adaptive method compared to state of the art competing methods (such as [9]).

[2]This result is produced by the software, available on http://decsai.ugr.es/∼javier/denoise/index.html.

REFERENCES

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., ser. Computer Science and Scientific Computing. Boston: Academic Press, 1990.

[2] M. P. Wand and M. C. Jones, *Kernel Smoothing, The Series of Monographis on Statistics and Applied Probability*. London; New York: Chapman and Hall, 1995.

[3] D. Ruppert and M. P. Wand, "Multivariate locally weighted least squares regression," *The Annals of Statistics*, vol. 22, no. 3, pp. 1346–1370, September 1994.

[4] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," submitted to *IEEE Transactions on Image Processing*, 2005.

[5] E. A. Nadaraya, "On estimating regression," *Theory of Probability and its Applications*, pp. 141–142, September 1964.

[6] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceeding of the 1998 IEEE International Conference of Compute Vision, Bombay, India*, pp. 836–846, January 1998.

[7] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1150, October 2002.

[8] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, November 2002.

[9] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, November 2003.

(a) Original image

(b) Noisy image, $\sigma = 25$

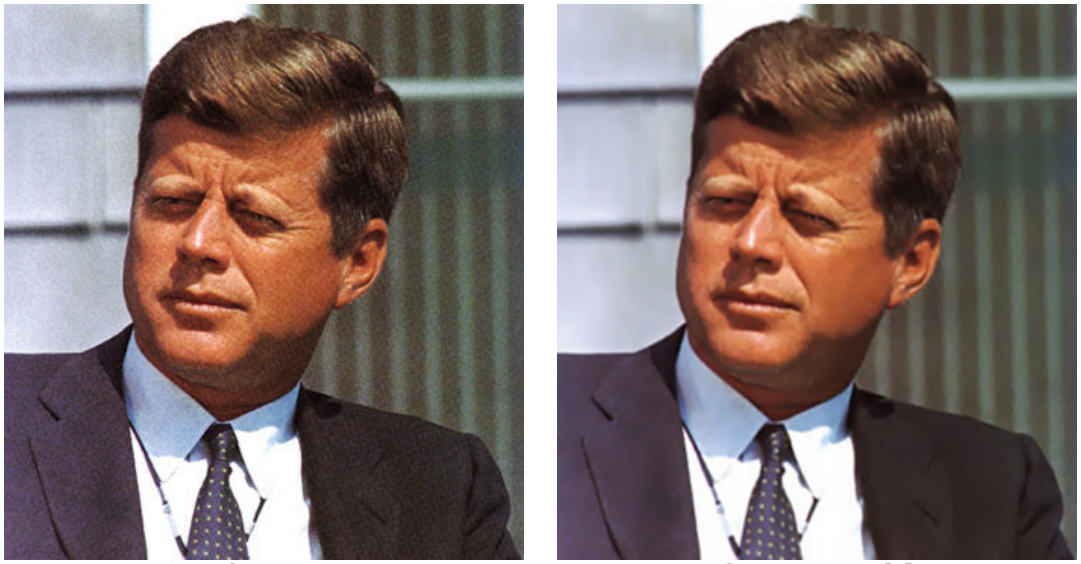
(c) BLS-GSM [9]

(d) Iterative steering kernel, $N = 2$

(e) Detail from (c)

(f) Detail from (d)

Fig. 4. The performance of different denoising methods are compared in this experiment. The RMSE of the images (b)-(d) are 25, 7.01, and 6.83, respectively. Gaussian kernel was used for all experiments.

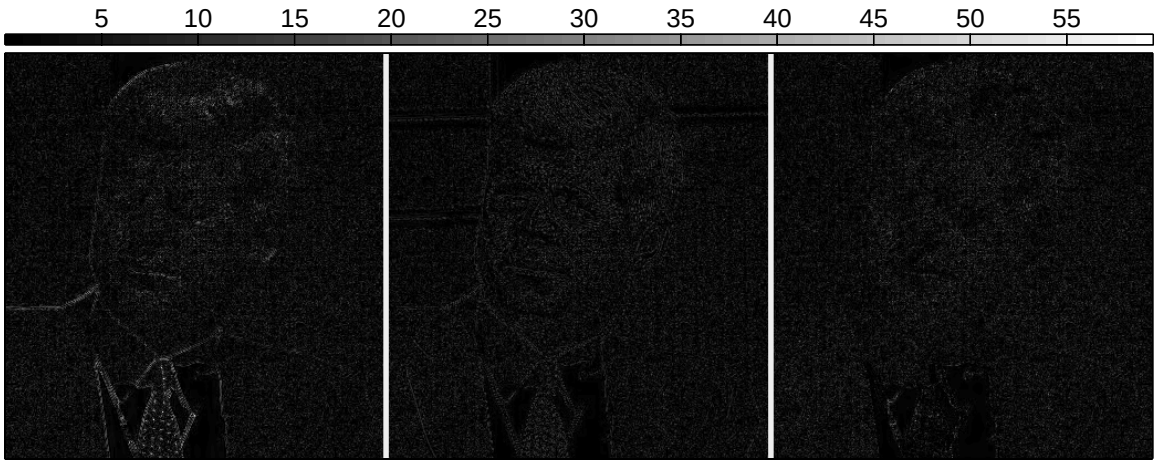


(a) Real noisy image

(b) BLS-GSM [9]

(c) Bilateral filter, $h = 2, h_r = 3.5$

(d) Iterative steering kernel, $N = 2, h = 2$



(e) BLS-GSM [9] (f) Bilateral filter (g) Iterative steering kernel

Fig. 5. The performance of different denoising methods are compared in this experiment on a color image with real noise. Gaussian kernel was used for all experiments. (e), (f) and (g) are absolute values of the residuals on the luminance channel.