Pattern Recognition 47 (2014) 1751–1763

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# A general non-local denoising model using multi-kernel-induced measures

Zhonggui Sun [a,b], Songcan Chen [a,*], Lishan Qiao [b]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, 210016 Nanjing, PR China
[b] Department of Mathematics Science, Liaocheng University, 252000 Liaocheng, PR China

ARTICLE INFO

*Article history:*
Received 2 August 2012
Received in revised form
26 October 2013
Accepted 4 November 2013
Available online 15 November 2013

*Keywords:*
Image denoising
Multi-kernel learning
Kernel-induced measure
Non-local means (NLM)
Complicated noise

ABSTRACT

Noises are inevitably introduced in digital image acquisition processes, and thus image denoising is still a hot research problem. Different from local methods operating on local regions of images, the non-local methods utilize non-local information (even the whole image) to accomplish image denoising. Due to their superior performance, the non-local methods have recently drawn more and more attention in the image denoising community. However, these methods generally do not work well in handling complicated noises with different levels and types. Inspired by the fact in machine learning field that multi-kernel methods are more robust and effective in tackling complex problems than single-kernel ones, we establish a general non-local denoising model based on multi-kernel-induced measures (GNLMKIM for short), which provides us a platform to analyze some existing and design new filters. With the help of GNLMKIM, we reinterpret two well-known non-local filters in the united view and extend them to their novel multi-kernel counterparts. The comprehensive experiments indicate that these novel filters achieve encouraging denoising results in both visual effect and PSNR index.



## 1. Introduction

Denoising is still a very active area of research in image processing, which goal is to reduce noise artifacts while retaining good details (such as edges) of observed images as much as possible [1]. Towards the end, many denoising methods, such as the mean filter, the total variation filter [2], the bilateral filter [3] and so on, were sequentially proposed and have obtained their popularity. Their common and effective characteristic is that they were all realized by locally averaging [4] which is an operation on a set of image values limited to a local targeted region in image. Though able to preserve image details to different degrees, the aforementioned local methods have two weaknesses: (1) the weights involved in the averaging only depend on the single pixels, thus sensitive to the polluted pixels; (2) artificial shocks (unpredictable artificial stripes or textures) usually appear in the denoised results [4].

To mitigate the weaknesses of local methods, Buades et al. proposed a non-local means filter (NLM) recently [4]. Unlike the local filters which typically operate on a *local* neighborhood, NLM operates on a *non-local* area (even the whole image) by using a dissimilarity measure between patches. Despite its simplicity and intuition in idea, NLM has been empirically validated to clearly outperform other classic filters including the aforementioned ones [5]. Inspired by such a patch-based and non-local viewpoint, many state-of-the-art non-local filters, including block-matching and three-dimensional filter (BM3D) [6], K-SVD [7], have been proposed in the recent years (refer to [5,8] for surveys in this topic).

In practice, the noise in an image is generally complicated, which can belong to different levels (strength) and different types (single or mixed) [9,10]. Although non-local methods have shown excellent performances in denoising, when faced with such complicated noises, they cannot necessarily be guaranteed to yield desirable denoising effects [11]. Take NLM as an example, it cannot keep high effectiveness in removing Gaussian noises with high levels, as shown in our experiment sections later. Also, as pointed out in [10], the NLM fails to remove the common mixed noise either. Therefore, the denoising ability of NLM still needs to be improved.

In terms of Huber robust statistics [12], robust filters are insensitive to outliers and can preserve image details well. Inspired by [12], Tan et al. developed a set of robust measures with the kernel-induced distances and then invented a more general filtering model (KIM for short) in [13], which not only motivates new robust filters to born but also accommodates the typical filters including the mean filter and the median filter as its special cases. Meanwhile, as a powerful learning paradigm in the machine learning community, it has been proved that multi-kernel methods are more flexible in handling complicated learning tasks than single-kernel ones [14]. For our current image denoising

* Corresponding author. Tel.: +86 25 84892956.
*E-mail address:* s.chen@nuaa.edu.cn (S. Chen).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.11.003

problem, the noises in images often have different levels and different types. Thus we conjecture that a filter designed by replacing single-kernel with multi-kernels can be more powerful in removing such complicated noises.

Motivated by such successful factors in NLM, KIM and multi-kernel methods, we develop a novel denoising model in this paper, which is general, non-local, based on multi-kernel-induced measures and thus named as GNLMKIM for short.

The main contributions of this paper are listed as follows:

(1) We propose a new denoising model (GNLMKIM), which provides a general platform for analyzing and designing filters. Besides the multi-kernel strategies, the novel model combines several favorable properties, such as robustness and non-locality of current effective filters. For such a proposal, we are not aware of similar works in literature.
(2) To illustrate the generality and effectiveness of GNLMKIM, we focus on two popular non-local filters including the NLM [4] and MNF (mixed noise filter) [10] as the starting point of our work and show that these two filters are both special cases of GNLMKIM.
(3) Based on GNLMKIM, we further extend both NLM and MNF to their multi-kernel counterparts with encouraging experimental results in removing wide-level (no matter low or high) Gaussian noises and mixed noises respectively.

The rest of this paper is structured as follows: In Section 2, we briefly review related works. Section 3 describes and solves GNLMKIM in detail. In Sections 4 and 5, we first reinterpret NLM and MNF as special cases of GNLMKIM, and then extend them to their multi-kernel counterparts followed with encouraging experimental results. Finally, we conclude this paper in Section 6.

## 2. Related works

Some related works including noise models, NLM, MNF, KIM and kernel strategy will be briefly introduced in turn in this section.

### 2.1. Introduction to noise models

Here we concern three main noise models corresponding to Gaussian noise, impulse noise and their mixture respectively. The Gaussian noise model is mathematically defined as

$$Y = X + N \tag{1}$$

where $Y$ is the observed (noisy) image, $X$ is the noise-free (original) one to be recovered, $N$ is the zero-mean Gaussian white noise with the standard deviation (STD) $\sigma$. A bounded domain on which $X$, $Y$, $N$ are defined is denoted as $\Omega = [1,\ldots,m]\times[1,\ldots,n]$.

There are mainly two common types of impulse noises, i.e., salt-and-pepper noise and random-valued noise. Let $N$ denote impulse noise. And suppose the gray values of an image are in the dynamic range $[N_{\min}, N_{\max}]$. For any pixel coordinate$i = (i_1, i_2) \in \Omega$, the two types of the impulse noises are respectively modeled as follows:

For the salt-and-pepper noise, we have

$$Y(i) = \begin{cases} N_{\min} & \text{with probabilty } s/2, \\ N_{\max} & \text{with probabilty } s/2, \\ X(i) & \text{with probabilty } 1-s. \end{cases} \tag{2}$$

where $0 \le s \le 1$ is the salt-and-pepper noise level.

For the random-valued impulse noise,

$$Y(i) = \begin{cases} N(i) & \text{with probabilty } r, \\ X(i) & \text{with probabilty } 1-r. \end{cases} \tag{3}$$

where $0 \le r \le 1$ is the noise level. The gray values, $N(i)$s, are identically and uniformly distributed random numbers in $[N_{\min}, N_{\max}]$. Throughout this paper, unless specified, the impulse noise means the random-valued type.

Finally the mixed noise model can be formulated as

$$Y = N_{imp}(X+N) \tag{4}$$

where $N_{imp}$ denotes an operator of image degradation with impulse noise.

### 2.2. Introduction to NLM

NLM is mainly designed for removing Gaussian noise. For any pixel coordinate $i = (i_1, i_2) \in \Omega$, NLM computes the restored (estimated) gray value $\hat{X}(i)$ as a weighted average in terms of

$$\hat{X}(i) = \frac{1}{C(i)} \sum_{j \in S_i} \exp\left(-\frac{||Y_i - Y_j||^2}{h^2}\right) Y(j) \tag{5}$$

where $S_i \subseteq \Omega$ is a non-local search window (even the whole $\Omega$ itself). Let $N^d\{i\} = \{(i_1+j_1, i_2+j_2) | (j_1, j_2) \in Z^2, |j_1| \le d, |j_2| \le d\}$. The restriction (patch) of $Y$ to the neighborhood of coordinate $i$ is defined as the vector $Y_i = Y(N^d\{i\}) = (Y(j) | j \in N^d\{i\})$. $h^2$ acts as a filtering parameter, which is recommended a value between $10\sigma$ and $15\sigma$ ($\sigma$ is the noise STD) [4,5,15]. $C(i) = \sum_{j \in S_i} \exp(||Y_i - Y_j||^2_{2,\alpha}/h^2)$ is a normalized factor. $||Y_i - Y_j||^2_{2,\alpha}$ is a Gaussian weighted dissimilarity measure (distance) between $Y_i$ and $Y_j$ and defined as

$$||Y_i - Y_j||^2_{2,\alpha} = \sum_{k \in K} G_\alpha(k)(Y(i-k) - Y(j-k))^2 \tag{6}$$

where $K = \{(k_1, k_2) | |k_1| \le d, |k_2| \le d\}$ is a local neighborhood centered at the origin. And

$$G_\alpha(k) = \frac{1}{2\pi\alpha^2} \exp\left(-\frac{{k_1}^2 + {k_2}^2}{2\alpha^2}\right), \quad k = (k_1, k_2) \tag{7}$$

is the Gaussian kernel with standard deviation $\alpha$.

Two distinct characteristics of NLM can be confirmed from the formulation in Eq. (5):

(1) The weights are based on the dissimilarities between patches.
(2) The restored gray value of each pixel $i$ is obtained from a non-local search window $S_i$. For the sake of computation, a $21 \times 21$ window centered in pixel $i$ is commonly recommended in [4,5,15].

### 2.3. Introduction to MNF

Due to totally different image degrading mechanisms brought out by Gaussian noise and impulse noise, up to date, just few works focus on the removal of the mixture of the two noises, though such a mixture is common in the real world. To make NLM capable to remove such mixed noise, Li et al. designed a mixed noise filter (MNF) [10] by incorporating rank-ordered absolute difference statistic (ROAD) into NLM, where the ROAD value is used to quantify how different in intensity the specific pixel is from its most similar neighbors (refer to [9] for more details). As a result, the output of MNF can be represented as

$$\hat{X}(i) = \frac{1}{C(i)} \sum_{j \in S_i} w_I(j) \exp\left(\frac{||Y_i - Y_j||^2_{2,M}}{2\sigma_M^2}\right) Y(j) \tag{8}$$

where $C(i)$ is still a normalized factor. $w_I(j)=\exp(-ROAD(j)^2/2\sigma_I^2)$ is the impulse weight. And $||Y_i-Y_j||^2_{2,M}$ is the impulse controlled weighted norm of $Y_i$ and $Y_j$, which is defined as

$$||Y_i-Y_j||^2_{2,M}=\sum_{k\in K} J_I(i-k,j-k)\,(Y(i-k)-Y(j-k))^2 \tag{9}$$

with $J_I(i,j)=\exp(-((ROAD(i)+ROAD(j))/2)^2/2\sigma_J^2)$. $\sigma_I$, $\sigma_J$ and $\sigma_M$ are controlling parameters in MNF. The other symbols in Eq. (8) have the same meanings as those in NLM introduced in Section 2.2. For the details of MNF, please refer to [10].

### 2.4. Introduction to kernel methods and KIM

Kernel methods are popular and important in machine learning [16] and have successfully motivated developing many learning methods such as SVM [17], Kernel Principal Component Analysis (KPCA) [18] and Kernel Fisher Discriminant (KFD) [19] to attack nonlinear classification, dimensionality-reduction and regression problems respectively. A common central idea behind these methods is kernel trick by which a nonlinear problem in an original low-dimensional input (data) space is mapped to a corresponding (approximately) linear problem in a higher dimensional feature space. An interesting point about the kernel trick is that the concrete form of the mapping need not be known as long as the involved operation between any pair of input data is the inner product [20].

With such kernel trick, Tan et al. induced nonlinear measures using some RBF kernels and then establish the kernel-induced measures model (KIM) [13] to design denoising algorithms. Concretely, let $\Phi$ be a mapping from a low-dimensional space to a higher-dimensional space, $X(i)$ and $Y(j)$ be any two pixels in the low-dimensional space, which belong to the original image $X$ and the noisy image $Y$ respectively, then the squared Euclidean distance in the higher-dimensional space corresponding to these two pixels can be expanded as

$$\begin{aligned}
&d(X(i),Y(j))^2=\|\Phi(X(i))-\Phi(Y(j))\|_2^2\\
&=(\Phi(X(i))-\Phi(Y(j)))^T(\Phi(X(i))-\Phi(Y(j)))\\
&=\langle\Phi(X(i)),\Phi(X(i))\rangle-2\langle\Phi(X(i)),\Phi(Y(j))\rangle+\langle\Phi(Y(j)),\Phi(Y(j))\rangle\\
&=K(X(i),X(i))-2K(X(i),Y(j))+K(Y(j),Y(j))
\end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product operation. Finally, KIM produces the following filtering output:

$$\hat{X}(i)=\arg\min_{X(i)}\sum_{j\in W} d(X(i),Y(j))^2 \tag{10}$$

where $W$ is a *local* sliding window centered at location $i$.

Different kernels generally induce different measures. For instance, when the $K$ is specified as Exponential (Gaussian) kernel, KIM proposed in Eq. (10) is accordingly specified as

$$\begin{aligned}
\hat{X}(i)&=\arg\min_{X(i)}\sum_{j\in W}(1-K(X(i),Y(j)))\\
&=\frac{\sum_{j\in W}K(X(i),Y(j))Y(j)}{\sum_{j\in W}K(X(i),Y(j))}
\end{aligned}$$

which is a nonlinear iteration solution to $X(i)$ and called Gaussian kernel based filter (GK) in [13].

By means of the KIM, the kernel trick can also be used to generalize the existing filtering algorithms such as the Mean filter and the Median filter to their kernelized counterparts to make them powerful in denoising [13]. Even so, KIM is still a pixel-based and local filter (model). From a viewpoint of machine learning, we can find that this model is just a paradigm using single kernel, e.g., Gaussian kernel. Inspired by the fact that multi-kernel methods are usually more flexible and powerful than single-kernel ones in handling complicated tasks in machine learning field [14] and motivated by the success of the two outstanding denoising methods (KIM and NLM), we develop a novel denoising model (GNLMKIM) in the next section. Besides combing the advantages of KIM and NLM, GNLMKIM inherits favorable properties of multi-kernel method and exhibits excellent performance in removing the complicated noises.

## 3. A non-local denoising model (GNLMKIM)

In this section, we first formulate the model and then give the corresponding solution.

### 3.1. Model formulation

The similarity between two image patches $X_i$ and $X_j$ can be expressed by a kernel $K(X_i,X_j)$. And similar to that in KIM, a robust dissimilarity measure between them can directly be induced by $1-K(X_i,X_j)$. Further, by means of multi-kernel method, we can also generalize the kernel $K$ to a multi-kernel which is linearly combined by $k$ base kernels. Now the new measure is turned into $d(X_i,X_j)^2=1-\sum_{t=1}^k\lambda_t K_t(X_i,X_j)$ with $\lambda_t\geq 0$ and $\sum_{t=1}^k\lambda_t=1$. Analogously to the classical weighted mean filter (WMF) which is derived by $\hat{X}(i)=\arg\min_{X(i)}\sum_{j\in W}w_j(X(i)-Y(j))^2$, we can also define GNLMKIM model as a weighted dissimilarity-minimized problem as follows:

$$\begin{aligned}
&\min_{X_i,\lambda}\quad\sum_{i\in\Omega}\sum_{j\in S_i}w_j\left(1-\sum_{t=1}^k\lambda_t K_t(X_i,Y_j)\right)-pH(\lambda)\\
&\text{s.t.}\quad\lambda_t\geq 0,\quad\sum_{t=1}^k\lambda_t=1.
\end{aligned}\tag{11}$$

And $\hat{X}(i)$, the central pixel of $\hat{X}_i$, is the filtering output of GNLMKIM. The $w_j$ in GNLMKIM plays the same role as that in WMF. Similar to that in NLM, $S_i$ is a search window by setting its size, GNLMKIM can operate in a non-local manner. Considering that the multiple kernel is a *linear* combination of many base kernels in Eq. (11), we need to introduce the regularization term $H(\lambda)$ with $\lambda=(\lambda_1,\lambda_2,\ldots,\lambda_k)^T$ to avoid yielding the degenerate solution. And $p$ is a nonnegative parameter controlling the degree of penalty for the $\lambda$.

In fact, the GNLMKIM model in Eq. (11) can equivalently be rewritten as

$$\begin{aligned}
&\min_{X_i,\lambda}\quad\sum_{i\in\Omega}\sum_{j\in S_i}w_j\sum_{t=1}^k(\lambda_t(1-K_t(X_i,Y_j)))-pH(\lambda)\\
&\text{s.t.}\quad\lambda_t\geq 0,\quad\sum_{t=1}^k\lambda_t=1.
\end{aligned}\tag{12}$$

As mentioned before, $1-K(\cdot,\cdot)$ is a dissimilarity measure. By comparing Eq. (11) with Eq. (12), we can interestingly find that the design problem of a filter in Eq. (11) in the *multi-kernel* fashion can be converted to an equivalent design problem in Eq. (12) in a *multi-measure* fashion. Specifically, on one hand, the objective in Eq. (11) is associated with *one* dissimilarity measure induced by the multi-kernel; on the other hand, the objective is associated with a linear combination of $k$ measures in Eq. (12). The different formulations remind us of the early design method of *combination filters* such as the mean-median (MEM) [21], mean-LogCauchy (MLC) [21] and mean-GK (MGK) [13] filters, which are designed by directly (linearly) combing two base *filters* (more precisely, their filtering results) instead of combing a set of *measures* in Eq. (12). And each such base filter is derived independently from its corresponding optimization objective associated with the Euclidian, Laplacian, Cauchy or Gaussian-kernel-induced measures [13]. That is, such an early combination filter is *NOT* resulted from

**Table 1**
Typical examples of entropy measure $H(\lambda)$.

| | | |
|---|---|---|
| (1) *Shannon entropy*: | $-\sum_t \lambda_t \log \lambda_t$ | Differentiable |
| (2) $l_2$-*entropy*: | $1-\lambda^T\lambda$ | Differentiable |
| (3) $l_1$-*entropy*: | $2-\sum_t \lvert\lambda_t-\frac{1}{n}\rvert$ | Non-differentiable |
| (4) $l_\infty$-*entropy*: | $1-\max\{\lambda_t \mid t=1,\ldots,k\}$ | Non-differentiable |

**Table 2**
Three commonly used kernels of $K_t$.

| | | |
|---|---|---|
| (1) *Exponential*: | $K(x,y)=\exp\left(-\frac{\lVert x-y\rVert_2^2}{h^2}\right)$ | Differentiable |
| (2) *Laplacian:* | $K(x,y)=\exp\left(-\frac{\lVert x-y\rVert_2^1}{h}\right)$ | Non-differentiable |
| (3) *Geman-McClure:* | $K(x,y)=1/\left(1+\frac{\lVert x-y\rVert_2^2}{h^2}\right)^2$ | Differentiable |

optimizing a single objective. However, the objective function of GNLMKIM in Eq. (12) is single and associated with the linearly combined measures. Thus, GNLMKIM provides us with a novel way to design filters. In this paper, we mainly focus on the multi-kernel viewpoint in Eq. (11) to discuss GNLMKIM and leave the designs based on the multi-measure viewpoint in Eq. (12) as our future work.

We highlight characteristics of GNLMKIM as follows:

(1) Its measures are patch-based and multi-kernel-induced.
(2) Its operation is non-local, similar to NLM.
(3) It is general. As illustrated later, the non-local filters such as the well-known NLM and the lately proposed MNF are both special cases of it and can be improved effectively.
(4) It is robust due to the case that a linear convex combination of robust kernels is still robust, for which we give a proof in Appendix A.

### 3.2. Model solution

From the constraint $\lambda_t \geq 0$ and $\sum_{t=1}^k \lambda_t = 1$ in Eq. (11), we find that the $\lambda_t$ s can be described in probability. To optimize it, a class of (discrete) entropies is often adopted for measuring the degree of uncertainty within a probability distribution. Table 1 lists typical examples of commonly used (generalized) entropy measures [22], any of which can be used to GNLMKIM as a prior constraint.

Considering the robustness kernels involved in Eq. (11) have different forms, we tabulate three commonly used kernels [13] in Table 2.

It should be mentioned here, for image denoising, when $\lVert x-y\rVert_2^2$ in Table 2 is replaced with $\lVert x-y\rVert_{2,\alpha}^2$ and $\lVert x-y\rVert_{2,M}^2$ respectively, the resulting $K(x,y)$s are still kernels.

In what follows, we derive GNLMKIM from Eq. (11) by using alternating minimization algorithm:

$$\text{Step 1}\quad \hat{\lambda}=\arg\min_{\lambda}\sum_{i\in\Omega}\sum_{j\in S_i} w_j\left(1-\sum_{t=1}^{k}\lambda_t K_t(Y_i,Y_j)\right)-pH(\lambda)$$
$$\text{s.t.}\quad \lambda_t\geq 0,\quad \sum\lambda_t=1 \tag{13}$$

$$\text{Step 2}\quad \hat{X}_i=\arg\min_{X_i}\sum_{i\in\Omega}\sum_{j\in S_i} w_j\left(1-\sum_{t=1}^{k}\hat{\lambda}_t K_t(X_i,Y_j)\right) \tag{14}$$

In step 1, we set $X_i=y_i$ to solve $\lambda$, and then in step 2, we fix $\lambda$ to solve $X_i$. Since the objective in Eq. (11) is not convex, our algorithm can only converge to a local optimum by iterating the two steps. Fortunately, based on a great deal of experiments, we find that one iteration is enough for obtaining promising results. Thus, we run the two steps only one time throughout all our experiments.

## 4. Extension of NLM

With the help of GNLMKIM, in this section, we reinterpret the well-known NLM and extend it to its multi-kernel counterpart.

### 4.1. NLM as a special case of GNLMKIM

Now we choose a single kernel $K(X_i,Y_j)=\exp\left(-\lVert X_i-Y_j\rVert_{2,\alpha}^2/h^2\right)$ and let all $w_j$s be equal in Eq. (11). As a result, GNLMKIM is reduced to

$$\hat{X}_i=\arg\min_{X_i}\sum_{j\in S_i}\left(1-\exp\left(-\frac{\lVert X_i-Y_j\rVert_{2,\alpha}^2}{h^2}\right)\right) \tag{15}$$

Let $D(G_\alpha)$ be an operator which returns a matrix with $G_\alpha$ on the main diagonal and the rest entries are all zeros. To solve the problem in Eq. (15), we zero the following derivative:

$$\frac{\partial\sum_{i\in\Omega}\sum_{j\in S_i}\left(1-\exp\left(-\frac{\lVert X_i-Y_j\rVert_{2,\alpha}^2}{h^2}\right)\right)}{\partial X_i} \tag{16}$$

for which we further have

$$\begin{aligned}
&=\frac{\partial\sum_{i\in\Omega}\sum_{j\in S_i}(1-\exp(-\lVert D(G_\alpha)(X_i-Y_j)\rVert_2^2/h^2))}{\partial X_i}\\
&=\frac{\partial\sum_{j\in S_i}(1-\exp(-(\sqrt{D(G_\alpha)}(X_i-Y_j))^T(\sqrt{D(G_\alpha)}(X_i-Y_j))/h^2))}{\partial X_i}\\
&=\sum_{j\in S_i}\left[-\frac{2(\sqrt{D(G_\alpha)}(X_i-Y_j))}{h^2}\exp\left(-\frac{\lVert(X_i-Y_j)\rVert_{2,\alpha}^2}{h^2}\right)\right]\\
&=0
\end{aligned}$$

As a result, a corresponding fixed point equation is got:

$$X_i=\frac{1}{C(i)}\sum_{j\in S_i}\exp\left(-\frac{\lVert X_i-Y_j\rVert_{2,\alpha}^2}{h^2}\right)Y_j \tag{17}$$

where $C(i)$ is still a normalized factor. Further, using $Y_i$ to substitute $X_i$ in $\lVert.\rVert$ of Eq. (17), we can obtain an approximate solution to the fixed point equation:

$$\hat{X}_i=\frac{1}{C(i)}\sum_{j\in S_i}\exp\left(-\frac{\lVert Y_i-Y_j\rVert_{2,\alpha}^2}{h^2}\right)Y_j$$

with

$$C(i)=\sum_{j\in S_i}\exp\left(-\frac{\lVert Y_i-Y_j\rVert_{2,\alpha}^2}{h^2}\right)$$

So, in this particular condition, $\hat{X}(i)=(1/C(i))\sum_{j\in S_i}\exp(-\lVert Y_i-Y_j\rVert_{2,\alpha}^2/h^2)Y(j)$ is the output of GNLMKIM and also the result of NLM. That is, NLM is a special case of GNLMKIM.

### 4.2. A multi-kernel extension of NLM

As mentioned before, different kernels have different characteristics for adapting different application environments. A kernel which effectively deals with low level noises is usually unsuitable for high level noises, and vice versa. In other words, using the single kernel in NLM does generally not work well in removing the wide (no matter low or high) level noises, which is also validated in our following experiments. To compensate the weakness of NLM in removing the complicated noises, we attempt to extend this filter to its multi-kernel counterpart based on GNLMKIM. Specifically, considering that the noises all belong to the same type (Gaussian noise), we choose homogenous kernels (two

Exponential kernels) to construct multi-kernel in GNLMKIM in Eq. (11) and set the $H(\lambda)$ as *Shannon entropy*. As shown in the following, this regularization term always leads to a closed-form solution regarding$\lambda$.

After setting up the parameters of GNLMKIM in Eq. (11), we specify the model as follows:

$$\min_{X_i,\lambda} \sum_{i\in\Omega}\sum_{j\in S_i}\left(1-\sum_{t=1}^{2}\lambda_t K_t(X_i,Y_j)\right)+p\sum_{t=1}^{2}(\lambda_t\ln(\lambda_t))$$
$$\text{s.t.}\quad \lambda_t\geq 0,\quad \sum_{t=1}^{k}\lambda_t=1. \tag{18}$$

with

$$K_1(X_i,Y_j)=\exp\left(-\frac{||X_i-Y_j||^2_{2,\alpha}}{{h_1}^2}\right)$$

$$K_2(X_i,Y_j)=\exp\left(-\frac{||X_i-Y_j||^2_{2,\alpha}}{{h_2}^2}\right)$$

Solving the problem in the two-step alternation described in Section 3.2, we get

$$\text{Step 1}\quad \hat{\lambda}=\arg\min_{\lambda}\sum_{i\in\Omega}\sum_{j\in S_i}\left(1-\sum_{t=1}^{2}\lambda_t K_t(Y_i,Y_j)\right)+p\sum_{t=1}^{2}(\lambda_t\ln(\lambda_t))$$
$$\text{s.t.}\quad \lambda_t\geq 0,\quad \sum\lambda_t=1 \tag{19}$$

$$\text{Step 2}\quad \hat{X}_i=\arg\min_{X_i}\sum_{i\in\Omega}\sum_{j\in S_i}\left(1-\sum_{t=1}^{2}\hat{\lambda}_t K_t(X_i,Y_j)\right) \tag{20}$$

To solve the subproblem in Eq. (19), we construct Lagrangian function as follows:

$$L(\lambda_t):=\sum_{i\in\Omega}\sum_{j\in S_i}\left(1-\sum_{t=1}^{2}\lambda_t K_t(Y_i,Y_j)\right)+p\sum_{t=1}^{2}\lambda_t\ln\lambda_t-\eta\left(\sum_{t=1}^{2}\lambda_t-1\right)$$

And let

$$\frac{\partial L}{\partial\lambda_t}=\sum_{i\in\Omega}\sum_{j\in S_i}-K_t(Y_i,Y_j)+p(\ln\lambda_t+1)-\eta=0$$

we have

$$\lambda_t=\exp\left(\frac{\eta}{p}-1\right)\exp\left(\frac{1}{p}\sum_{i\in\Omega}\sum_{j\in S_i}K_t(Y_i,Y_j)\right)$$

Since $\sum\lambda_t=1$

$$\exp\left(\frac{\eta}{p}-1\right)=\frac{1}{\sum_{t=1}^{2}\exp\left(\frac{1}{p}\sum_{i\in\Omega}\sum_{j\in S_i}(K_t(Y_i,Y_j))\right)}$$

Then, we obtain the solution to the subproblem in Eq. (19):

$$\hat{\lambda}_t=\frac{1}{U(t)}\exp\left(\frac{1}{p}\sum_{i\in\Omega}\sum_{j\in S_i}K_t(Y_i,Y_j)\right)$$

where

$$U(t)=\sum_{t=1}^{2}\exp\left(\frac{1}{p}\sum_{i\in\Omega}\sum_{j\in S_i}K_t(Y_i,Y_j)\right)$$

To solve the subproblem in Eq. (20), we solve

$$\frac{\partial\sum_{i\in\Omega}\sum_{j\in S_i}(1-\sum_{t=1}^{2}\hat{\lambda}_t K_t(X_i,Y_j))}{\partial X_i}=0$$

similar to the method used in Section 4.1, and get

$$\hat{X}_i=\frac{1}{C(i)}\sum_{j\in S_i}\sum_{t=1}^{2}(\hat{\lambda}_t K_t^i(Y_i,Y_j))Y_j$$

with

$$C(i)=\sum_{j\in S_i}\sum_{t=1}^{2}(\hat{\lambda}_t K_t'(Y_i,Y_j))$$

and

$$K_t'(Y_i,Y_j)=\frac{1}{{h_t}^2}K_t(Y_i,Y_j)$$

Finally, the filtering output of the novel multi-kernel filter becomes

$$\hat{X}(i)=\frac{1}{C(i)}\sum_{j\in S_i}\sum_{t=1}^{2}(\hat{\lambda}_t K_t'(Y_i,Y_j))Y(j) \tag{21}$$

From the above discussion, we can find that NLM and its multi-kernel counterpart we proposed here both need computing values of the kernel functions, which is expensive in practice. Concretely, for an image of size $M\times M$, the computational complexity of NLM is $O(M^4(d\times d))=O(M^4d^2)$, and that of the multi-kernel version is $O(2M^4d^2)=O(M^4d^2)$. Here, $d\times d$ is the size of the patches.

### 4.3. Experimental results

All experiments are conducted on Core2 Duo 2.2 GHz (2 GB memory). And the software environment is Matlab 7.11. Throughout the experiments, we mainly focus on validating the effectiveness of the proposed multi-kernel extensions. Speeding up [23–25], threshold setting [26] and other improving strategies for NLM go beyond the scope of this paper. Similar to single-kernel methods, the strategies [23–26] can also be borrowed to boost our multi-kernel methods.

Different parameter settings of a kernel or filter often lead to different denoising effects. To objectively conduct the experiments, we set all the parameters of the compared methods to the values recommended in their corresponding original documents respectively unless specified.

(1) Three single exponential kernels are tested in our experiments. In the first one, the kernel parameter $h^2=10\sigma$ which is the setting recommend in [4,15] for NLM. In the second one, $h^2=100\sigma$, and in the last one, $h^2=1000\sigma$.
(2) In our method, we choose the multi-kernel composed of the first two single kernels and set the regularization parameter $p=5000$. Although the setting for the parameters of our multi-kernel method is not necessarily the best, it has shown promising results in the following experiments.
(3) By adding the single kernel with $h^2=1000\sigma$ to our method, we get another multi-kernel (K3 for short) which is combined by all the three single ones.
(4) For all the filters involved in the experiment, as recommended in [4,5,15], the size of patches and search windows are set to $7\times 7$ and $21\times 21$ respectively.

In our experiments, we select six frequently used images with different degrees of detail as shown in Fig. 1.

As usual, we evaluate the quality of restored images in terms of visual effect and quantitative index of peak signal-to-noise-ratio (PSNR) which is defined as

$$PSNR=10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{22}$$

with $MSE=(1/|\Omega|)\sum_{i\in\Omega}(X(i)-\hat{X}(i))^2$. According to the definition, the larger the PSNR value, the better the restored quality is. First, we list the specific PSNR results on the tested images in Table 3.

Then, in Fig. 2, we illustrate the performances of the compared filters in terms of average PSNR corresponding to Table 3.


**Fig. 1.** Six images ($512 \times 512$) for denoising experiment

**Table 3**
The PSNR (dB) results with Gaussian noise. The numbers in bold are the largest PSNR values corresponding to the results of the different denoising methods.

| | Noise STD | $\sigma=10$ | $\sigma=30$ | $\sigma=50$ | $\sigma=70$ | $\sigma=90$ | $\sigma=100$ |
|---|---|---|---|---|---|---|---|
| Lenna | $h^2=10\sigma$ (NLM)[4] | **34.81** | 27.33 | 21.47 | 17.87 | 15.67 | 14.90 |
| | $h^2=100\sigma$ | 29.88 | 27.16 | 25.44 | **23.84** | **22.39** | **21.76** |
| | $h^2=1000\sigma$ | 24.65 | 23.17 | 22.57 | 21.94 | 21.20 | 20.82 |
| | Ours | 34.73 | **27.39** | **25.55** | 23.81 | 22.30 | 21.70 |
| | K3 | 34.73 | **27.39** | **25.55** | 23.81 | 22.30 | 21.71 |
| Boats | $h^2=10\sigma$ (NLM)[4] | **32.58** | 26.40 | 21.07 | 17.58 | 15.45 | 14.71 |
| | $h^2=100\sigma$ | 27.18 | 24.61 | 23.30 | 22.16 | **21.15** | **20.70** |
| | $h^2=1000\sigma$ | 22.58 | 21.52 | 21.09 | 20.67 | 20.17 | 19.91 |
| | Ours | 32.49 | **26.43** | **23.43** | **22.20** | 21.11 | 20.63 |
| | K3 | 32.49 | **26.43** | **23.43** | **22.20** | 21.11 | 20.63 |
| Bridge | $h^2=10\sigma$ (NLM)[4] | **29.06** | **24.54** | 20.47 | 17.47 | 15.45 | 14.72 |
| | $h^2=100\sigma$ | 23.98 | 22.07 | 21.11 | 20.26 | **19.42** | **19.02** |
| | $h^2=1000\sigma$ | 20.48 | 19.61 | 19.26 | 18.89 | 18.44 | 18.19 |
| | Ours | 29.02 | 24.52 | **21.23** | **20.31** | 19.39 | 19.00 |
| | K3 | 29.02 | 24.52 | **21.23** | **20.31** | 19.39 | 19.00 |
| Barbara | $h^2=10\sigma$ (NLM)[4] | **33.75** | 26.45 | 21.19 | 17.81 | 15.65 | 14.88 |
| | $h^2=100\sigma$ | 27.60 | 24.27 | 22.70 | 21.43 | **20.32** | **19.82** |
| | $h^2=1000\sigma$ | 21.92 | 20.81 | 20.37 | 19.88 | 19.30 | 18.99 |
| | Ours | 33.61 | **26.51** | **22.86** | **21.49** | 20.29 | 19.77 |
| | K3 | 33.61 | **26.51** | **22.86** | **21.49** | 20.29 | 19.79 |
| Peppers | $h^2=10\sigma$ (NLM)[4] | **34.26** | 27.39 | 21.59 | 18.00 | 15.77 | 14.99 |
| | $h^2=100\sigma$ | 30.30 | 27.20 | 25.03 | **23.18** | **21.60** | **20.93** |
| | $h^2=1000\sigma$ | 24.43 | 22.49 | 21.73 | 21.01 | 20.23 | 19.84 |
| | Ours | 34.17 | **27.43** | **25.43** | 23.12 | 21.53 | 20.89 |
| | K3 | 34.17 | **27.43** | **25.43** | 23.12 | 21.53 | 20.89 |
| Couple | $h^2=10\sigma$ (NLM)[4] | **32.43** | 26.08 | 20.86 | 17.44 | 15.36 | 14.65 |
| | $h^2=100\sigma$ | 26.32 | 23.98 | 22.88 | 21.96 | **21.12** | **20.74** |
| | $h^2=1000\sigma$ | 22.24 | 21.39 | 21.07 | 20.73 | 20.32 | 20.09 |
| | Ours | 32.36 | **26.21** | **23.02** | **22.03** | 21.07 | 20.66 |
| | K3 | 32.36 | **26.21** | **23.02** | **22.03** | 21.07 | 20.69 |

From the results above, we can observe that

(1) Different kernels generally adapt different noises. The single kernel with $h^2=10\sigma$ can effectively deal with low level noises and gradually tends to fail as the level is increased higher than 50, while the single kernel with $h^2=100\sigma$ is only active for the high level noises. However, the single kernel with $h^2=1000\sigma$ hardly remove any level noise added in this experiment, implying that it is not appropriate (nonsensical) in such a case.

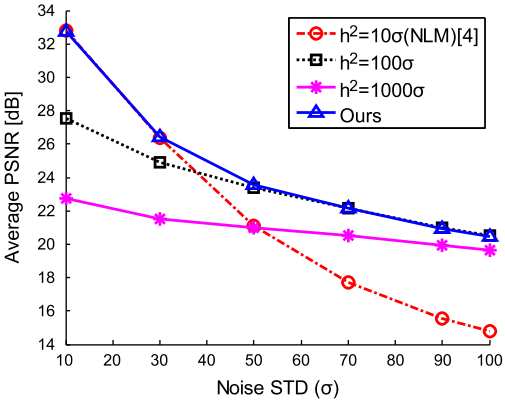


**Fig. 2.** Comparison in terms of average PSNR corresponding to Table 3.

(2) Our multi-kernel method has more flexibility by optimizing the linear combination of the two kernels, which can well remove wide (no matter low or high) level noises;
(3) From the behaviors of K3 filter, we notice that adding a nonsensical kernel does not produce unfavorable denoising effect.

In addition to the above observations, it is worthwhile to point out that our multi-kernel filtering method cannot necessarily always outperform the single kernels in attacking individual noise level (except for 50), which is probably because that an appropriately chosen single kernel can be affordable enough to remove such just-one-level noise.

Finally, we show a few closer investigations of the noisy images and the corresponding denoised ones in Figs. 3 and 4.

The visual effects above also show that the extension of NLM based on multi-kernel method is excellent in removing Gaussian noise. Compared with the three single-kernel ones, the novel filter has better behaviors in the *wide* noise levels.

## 5. Extension of MNF

To further illustrate the effectiveness of GNLMKIM, based on this model, we extend the lately proposed mixed noise filter (MNF) [10] to its multi-kernel counterpart too.

### 5.1. A multi-kernel extension

Similar to the extension of NLM, we give the reinterpretation of MNF first. Choosing a single kernel $K(X_i,Y_j)=\exp(-||X_i-Y_j||^2_{2,M}/2\sigma_M^2)$ and replacing $w_j$ with $w_I(j)$ (proposed in Eq. (8)), we specify GNLMKIM model in Eq. (11) into the following formulation:

$$\hat{X}_i=\arg\min_{X_i}\sum_{j\in S_i}w_I(j)\left(1-\exp\left(-\frac{||X_i-Y_j||^2_{2,M}}{2{\sigma_M}^2}\right)\right) \tag{23}$$

Analogously solving the problem in Eq. (15) in Section 4.1, we get $\hat{X}(i)=(1/C(i))\sum_{j\in S_i}w_I(j)\exp(-||Y_i-Y_j||^2_{2,M}/2{\sigma_M}^2)Y(j)$ which is the denoised output of MNF in Eq. (8) indeed. That is, MNF is also a special case of GNLMKIM.

Considering that MNF is mainly designed for removing the mixture of Gaussian and impulse noises, and in order to improve the denoising ability of this filter, we select two heterogeneous kernels (Exponential kernel and Geman-McClure kernel) to construct the multi-kernel in Eq. (11). In fact, richer and more types of kernels can be used in practical conditions. Here, we still use *Shannon entropy* to regularize the $\lambda$. As a result, GNLMKIM is specified as follows:

$$\min_{X_i,\lambda}\ \sum_{i\in\Omega}\sum_{j\in S_i}w_I(j)\left(1-\sum_{t=1}^{2}\lambda_t K_t(X_i,Y_j)\right)+p\sum_{t=1}^{2}(\lambda_t\ln(\lambda_t))$$
$$\text{s.t.}\quad \lambda_t\geq 0,\quad \sum_{t=1}^{k}\lambda_t=1 \tag{24}$$

where

$$K_1(X_i,Y_j)=\exp\left(-\frac{||X_i-Y_j||^2_{2,M}}{{h_1}^2}\right)$$

and

$$K_2(X_i,Y_j)=1/\left(1+\frac{||X_i-Y_j||^2_{2,M}}{h_2^2}\right)^2$$

According to the two-step alternation method described in Section 3.2, we get

$$\text{Step 1}:\ \hat{\lambda}=\arg\min_{\lambda}\sum_{i\in\Omega}\sum_{j\in S_i}w_I(j)\left(1-\sum_{t=1}^{2}\lambda_t K_t(Y_i,Y_j)\right)+p\sum_{t=1}^{2}(\lambda_t\ln(\lambda_t))$$
$$\text{s.t.}\quad \lambda_t\geq 0,\ \sum\lambda_t=1 \tag{25}$$

$$\text{Step 2}:\quad \hat{X}_i=\arg\min_{X_i}\sum_{i\in\Omega}\sum_{j\in\Omega}w_I(j)\left(1-\sum_{t=1}^{2}\hat{\lambda}_t K_t(X_i,Y_j)\right) \tag{26}$$

Similar to the methods in Section 4.2, we solve these two subproblems and get the output of the novel filter, expressed by

$$\hat{X}=(i)=\frac{1}{C(i)}\sum_{j\in S_i}\sum_{t=1}^{2}(\hat{\lambda}_t w_I(j)K_t'(Y_i,Y_j))Y(j) \tag{27}$$

where $C(i)$ is still a normalized factor

$$K_1'(Y_i,Y_j)=\frac{1}{h_1^2}\exp\left(-\frac{||Y_i-Y_j||^2_{2,M}}{{h_1}^2}\right)$$
$$K_2'(Y_i,Y_j)=\frac{2}{h_2^2}/\left(1+\frac{\|X_i-Y_j\|^2_{2,M}}{{h_2}^2}\right)^3$$

and

$$\hat{\lambda}_t=\frac{1}{U(t)}\exp\left(\frac{1}{p_i}\sum_{i\in\Omega}\sum_{j\in S_i}w_I(j)K_t(Y_i,Y_j)\right)$$

with

$$U(t)=\sum_{t=1}^{2}\exp\left(\frac{1}{p_i}\sum_{i\in\Omega}\sum_{j\in S_i}w_I(j)K_t(Y_i,Y_j)\right)$$

### 5.2. Experimental results

To illustrate the effectives of the multi-kernel modification, we compare our novel filter with the original MNF in denoising. Two classical filters, signal-dependent rank ordered mean (SD-ROM) filter [27] and Trilateral filter [9], for removing the mixed noised are also considered here.

To objectively conduct the experiments, we still set the parameters of the compared filters to the values recommended in their corresponding original documents. Similar to NLM, in our novel filter, we believe $h_i^2(i=1,2)$ should also depend on noise level and thus directly set ${h_i}^2=a_i\sigma+b_i r$ (where $a_i$ and $b_i$ are parameters to be settled, $\sigma$ is still the Gaussian noise STD, and $r$ is the impulse noise level defined in Eq.(3)). In fact, the 1st term of the sum results from the influence of Gaussian noise while the 2nd term is from the influence of impulse noise. In such setting, we find that good selections for $h_1$ and $h_2$ are $h_1^2=5\sigma+150r$ and ${h_2}^2=\sigma+30r$,


**Fig. 3.** Closer investigations of the six tested images corrupted by Gaussian noise with $\sigma=30$. From left to right: (a) the noisy data, (b–d) restored by the single kernel methods with $h^2=10\sigma$ (NLM)[4], $100\sigma$, $1000\sigma$ respectively, and (e) restored by our multi-kernel method.

respectively. And the other parameters are set the same values as those in MNF. Specifically, as recommended in [10], $\sigma_I=\sigma_J=100+\sigma-160r$, the size of patches and search windows are set to $3\times3$ and $7\times7$ respectively. Although the above parameter settings for our novel filter are not necessarily the best, their corresponding experimental results are promising.

Next, we report the corresponding PSNR in Table 4.

To compare their visual effects, we also show a few denoising closer investigations in Fig. 5.

Jointly from Table 4 and Fig. 5, we can find that, similar to the extension of NLM in Section 4, the multi-kernel counterpart of MNF is very effective too. And its filtering results in removing the mixed noise defeat the original mixed noise filter, i.e., MNF, in both visual effect and PSNR.

Further, in Table 5 and Fig. 6, we give the experimental results on removing the wide level Gaussian noises.

In removing the Gaussian noise, as showed in the results above, our multi-kernel method also exhibits effectiveness, which


**Fig. 4.** Closer investigations of the six tested images corrupted by Gaussian noise with $\sigma=70$. From left to right: (a) the noisy data, (b–d) restored by the single kernel methods with $h^2=10\sigma$ (NLM) [4], $100\sigma$, $1000\sigma$ respectively, and (e) restored by our multi-kernel method.

outperforms the SD-ROM filter and the Trilateral filter apparently. We also notice that, compared with the original MNF, the multi-kernel method is not so effective when the noise level is quite high ($\sigma > 60$). That is, selecting the two heterogeneous kernels (Exponential kernel and Geman-McClure kernel) to construct the multi-kernel does not necessarily always work well in removing the homogenous noise, e.g., Gaussian noise. Thus, in specific applications, the kernel members involved in the combination need to be selected as in terms of available prior knowledge as possible.

## 6. Conclusions

In this paper, inspired by the multi-kernel learning, we propose a general non-local denoising model incorporating multi-kernel-induced measures (GNLMKIM). This model is robust and patch-based. As a general model, GNLMKIM provides us a platform to analyze and design non-local filters. With the help of the platform, we not only reinterpret the two typical non-local filters (NLM and MNF), but also extend them to their multi-kernel counterparts with encouraging experimental results.

**Table 4**
The PSNR (dB) results on the tested images with the mixed noise. The numbers in bold are the largest PSNR values corresponding to the results of the different denoising methods.

| | | $r=0.1$ | | | | $r=0.2$ | | | | $r=0.3$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\sigma=15$ | $\sigma=20$ | $\sigma=25$ | $\sigma=30$ | $\sigma=15$ | $\sigma=20$ | $\sigma=25$ | $\sigma=30$ | $\sigma=15$ | $\sigma=20$ | $\sigma=25$ | $\sigma=30$ |
| Lenna | SD-ROM [27] | 25.90 | 24.23 | 22.99 | 22.08 | 25.51 | 23.83 | 22.67 | 21.74 | 24.75 | 23.25 | 22.16 | 21.31 |
| | Trilateral [9] | 29.72 | 28.39 | 27.11 | 25.80 | 28.64 | 27.42 | 26.21 | 24.98 | 27.40 | 27.05 | 26.59 | 26.11 |
| | MNF [10] | 30.02 | 29.30 | 28.69 | 28.02 | 29.48 | 28.82 | 28.10 | 27.43 | 28.85 | 28.20 | 27.49 | **26.76** |
| | Ours | **31.60** | **30.49** | **29.53** | **28.48** | **30.80** | **29.74** | **28.69** | **27.66** | **29.70** | **28.69** | **27.62** | 26.62 |
| Boats | SD-ROM [27] | 25.19 | 23.65 | 22.53 | 21.65 | 24.64 | 23.20 | 22.12 | 21.31 | 23.90 | 22.61 | 21.62 | 20.80 |
| | Trilateral [9] | 27.13 | 26.34 | 25.44 | 24.45 | 26.21 | 25.43 | 24.57 | 23.69 | 24.75 | 24.53 | 24.17 | 23.88 |
| | MNF [10] | 26.81 | 26.29 | 25.84 | 25.42 | 26.32 | 25.82 | 25.35 | 24.95 | 25.80 | 25.35 | 24.90 | 24.45 |
| | Ours | **28.73** | **27.92** | **27.19** | **26.44** | **27.88** | **27.14** | **26.36** | **25.73** | **26.97** | **26.26** | **25.55** | **24.83** |
| Bridge | SD-ROM [27] | 24.05 | 22.77 | 21.75 | 20.96 | 23.37 | 22.21 | 21.27 | 20.54 | 22.47 | 21.51 | 20.67 | 19.96 |
| | Trilateral [9] | 24.51 | 23.94 | 23.30 | 22.67 | 23.70 | 23.18 | 22.58 | 21.91 | 22.28 | 22.15 | 21.92 | 21.68 |
| | MNF [10] | 23.81 | 23.50 | 23.20 | 22.95 | 23.49 | 23.17 | 22.88 | 22.60 | 23.10 | 22.84 | 22.53 | 22.23 |
| | Ours | **25.30** | **24.80** | **24.31** | **23.88** | **24.73** | **24.24** | **23.78** | **23.33** | **24.01** | **23.60** | **23.14** | **22.64** |
| Barbara | SD-ROM [27] | 23.24 | 22.22 | 21.38 | 20.70 | 22.76 | 21.80 | 21.00 | 20.32 | 22.13 | 21.29 | 20.53 | 19.88 |
| | Trilateral [9] | 24.13 | 23.62 | 23.12 | 22.51 | 23.53 | 23.08 | 22.59 | 21.96 | 22.70 | 22.56 | 22.38 | 22.19 |
| | MNF [10] | 24.81 | 24.27 | 23.86 | 23.52 | 24.18 | 23.77 | 23.45 | 23.13 | 23.62 | 23.31 | 23.03 | 22.74 |
| | Ours | **27.19** | **26.16** | **25.31** | **24.59** | **25.79** | **25.03** | **24.40** | **23.79** | **24.51** | **23.99** | **23.46** | **22.97** |
| Peppers | SD-ROM [27] | 25.80 | 24.10 | 22.91 | 22.02 | 25.24 | 23.69 | 22.52 | 21.65 | 24.41 | 22.99 | 21.94 | 21.04 |
| | Trilateral [9] | 28.10 | 27.16 | 26.17 | 25.15 | 27.34 | 26.50 | 25.43 | 24.41 | 26.53 | 26.23 | 25.81 | 25.29 |
| | MNF [10] | 30.26 | 29.50 | 28.70 | 28.05 | 29.49 | 28.76 | 27.91 | 27.31 | 28.64 | 27.90 | 27.16 | **26.37** |
| | Ours | **31.55** | **30.60** | **29.55** | **28.59** | **30.56** | **29.63** | **28.51** | **27.66** | **29.25** | **28.32** | **27.34** | 26.25 |
| Couple | SD-ROM [27] | 25.14 | 23.59 | 22.50 | 21.60 | 24.56 | 23.18 | 22.08 | 21.26 | 23.82 | 22.55 | 21.59 | 20.76 |
| | Trilateral [9] | 26.81 | 25.98 | 25.10 | 24.17 | 25.92 | 25.20 | 24.37 | 23.49 | 24.38 | 24.25 | 23.98 | 23.63 |
| | MNF [10] | 26.19 | 25.72 | 25.32 | 24.94 | 25.77 | 25.36 | 24.93 | 24.57 | 25.31 | 24.96 | 24.51 | 24.13 |
| | Ours | **28.19** | **27.32** | **26.64** | **25.94** | **27.39** | **26.68** | **25.99** | **25.32** | **26.52** | **25.94** | **25.20** | **24.56** |

To conclude, two possible extensions of the multi-kernel methods derived from GNLMKIM are as follows:

(1) Similar to NLM, the multi-kernel methods are generally complex in computation. Fortunately, as mentioned before, there have been some fast algorithms designed [23–25] for speeding-up NLM. It is possibly feasible to borrow the strategies to speed up the multi-kernel methods.
(2) Constructed by two individual kernel members, the multi-kernel methods have already exhibited excellent ability to remove the complicated noises in this paper. However, as illustrated in the experiments, there is still room for further boosting the denoising performance. How to adjust the numbers and types of the combined members to specific applications is a meaningful yet challenging work, and is deserved as a future study.

## Conflict of interest statement

None declared.

## Acknowledgments

The authors would like to thank B. Li, and X. Luo et al. for offering their codes. And this work was partly supported by National Natural Science Foundations of China (#61170151, #61300154), Natural Science Foundation of Shandong Province (ZR2010FL011, ZR2012FQ005), Fundamental Research Funds for the Central Universities (NZ2013306) and sponsored by Jiangsu Qing Lan Project.

## Appendix A

Proof that GNLMKIM is a robust estimator

There are many robust estimators, e.g., M-estimator, L-estimator and R-estimator [12]. Here we only focus on M-estimator. From Eq. (14), we know $\{Y_j j \in S_i\}$ is an observed dataset and $X_i$ is an unknown parameter to be estimated. Then, the M-estimator for the location estimation can be generated by minimizing the following function:

$$J(X_i)=\sum_{i\in\Omega j\in S_i}\sum\rho(X_i-Y_j) \tag{A.1}$$

where

$$\rho(X_i-Y_j)=w_j\left(1-\sum_{t=1}^{k}\lambda_t K_t(X_i,Y_j)\right)$$

is a function which only depend on $X_i-Y_j$ in this proof. Then the M-estimator is generated by solving the following equation:

$$J'(X_i)=\sum_{j\in S_i}\phi(X_i-Y_j)=0 \tag{A.2}$$

where $\phi(X_i-Y_j)=(\partial/\partial X_i)\rho(X_i-Y_j)$. The influence function (*IF*) of M-estimator [12] which is proportional to $\phi$ and defined as

$$IF(Y_j;F,X_i)=\frac{\phi(X_i-Y_j)}{\int\phi'(X_i-Y_j)dF_{Y_j}(Y_j)} \tag{A.3}$$

where $F_{Y_j}(Y_j)$ is the distribution function of $Y_j$. If the *IF* of an estimator is unbounded, an outlier might cause trouble, i.e., the estimator is not robust. There are many measures to estimate the robustness derived from *IF*. And an important one of them is the gross error sensitivity (GES), which is defined as

$$\gamma^*=\sup_{Y_j}|IF(Y_j;F,X_i)| \tag{A.4}$$

where $|*|$ denotes *ABS* function (for a scalar) or $l_2$ norm (for a vector). The value of GES can interpret the worst approximate influence that the addition of an infinitesimal point mass can have on the value of the associated estimator. For simplicity, here we only use two single kernels (Exponential and Geman-McClure which definitions can be found in Table 2) to construct the multiple kernel in the function $\rho$ in Eq. (A.1). For other forms of the multi-kernel, it can be proved by taking similar tactics. Thus,

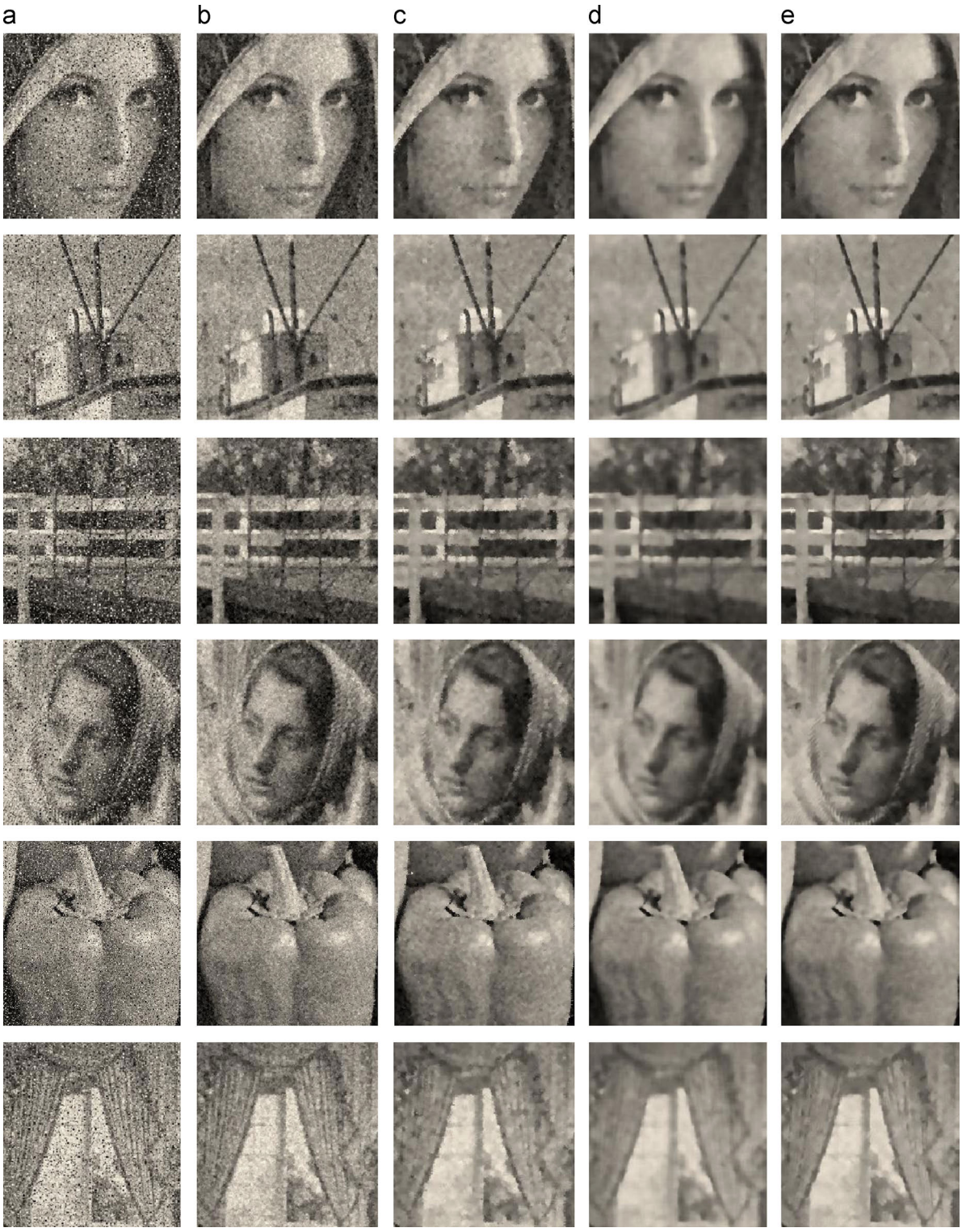


**Fig. 5.** Closer investigations of the three tested images corrupted by the mixed noise with $r=0.2$ and $\sigma=25$. From left to right: (a) the noisy data, (b) restored by SD-ROM [27], (c) restored by Trilateral [9], (d) restored by MNF [10] and (e) restored by our multi-kernel method.

we have

$$\rho = w_j(1-\lambda_1 K_1(X_i, Y_j)-\lambda_2 K_2(X_i, Y_j)) \tag{A.5}$$

with

$$K_1(X_i, Y_j) = \exp\left(-\frac{||X_i - Y_j||_2^2}{{h_1}^2}\right)$$

and

$$K_2(X_i, Y_j) = 1/\left(1+\frac{||X_i - Y_j||_2^2}{{h_2}^2}\right)^2$$

Therefore

$$|\phi(X_i - Y_j)|$$

**Table 5**
The PSNR (dB) results with Gaussian noise. The numbers in bold are the largest PSNR values corresponding to the results of the different denoising methods.

| Noise STD | | $\sigma=10$ | $\sigma=30$ | $\sigma=50$ | $\sigma=70$ | $\sigma=90$ | $\sigma=100$ |
|---|---|---|---|---|---|---|---|
| Lenna | SD-ROM [27] | 28.86 | 22.31 | 20.14 | 19.00 | 18.10 | 17.83 |
| | Trilateral [9] | 32.46 | 26.57 | 21.90 | 18.52 | 15.90 | 14.85 |
| | MNF [10] | 31.39 | 28.51 | **26.33** | **24.37** | **22.42** | **21.63** |
| | Ours | **33.56** | **29.13** | 25.64 | 22.72 | 20.14 | 19.13 |
| Boats | SD-ROM [27] | 27.88 | 21.91 | 19.80 | 18.61 | 17.80 | 17.52 |
| | Trilateral [9] | 29.06 | 25.13 | 21.32 | 18.10 | 15.74 | 14.78 |
| | MNF [10] | 28.14 | 25.80 | 24.27 | **22.89** | **21.41** | **20.73** |
| | Ours | **30.68** | **27.03** | **24.32** | 21.84 | 19.60 | 18.67 |
| Bridge | SD-ROM [27] | 26.52 | 21.28 | 19.27 | 18.11 | 17.30 | 17.04 |
| | Trilateral [9] | 26.00 | 23.32 | 20.41 | 17.63 | 15.44 | 14.67 |
| | MNF [10] | 24.71 | 23.23 | 22.26 | **21.30** | **20.25** | **19.69** |
| | Ours | **26.62** | **24.32** | **22.54** | 20.70 | 18.89 | 18.02 |
| Barbara | SD-ROM [27] | 25.09 | 20.95 | 19.15 | 18.09 | 17.41 | 17.07 |
| | Trilateral [9] | 25.31 | 23.03 | 20.32 | 17.68 | 15.59 | 14.72 |
| | MNF [10] | 26.97 | 23.94 | 22.79 | **21.76** | **20.61** | **19.98** |
| | Ours | **30.53** | **25.47** | **22.99** | 20.99 | 19.11 | 18.14 |
| Peppers | SD-ROM [27] | 28.84 | 22.32 | 20.13 | 18.92 | 18.05 | 17.68 |
| | Trilateral [9] | 29.51 | 25.84 | 21.84 | 18.39 | 15.99 | 14.95 |
| | MNF [10] | 32.06 | 28.79 | **26.31** | **24.22** | **22.20** | **21.42** |
| | Ours | **33.58** | **29.40** | 25.71 | 22.61 | 19.98 | 18.94 |
| Couple | SD-ROM [27] | 27.85 | 21.87 | 19.76 | 18.63 | 17.84 | 17.41 |
| | Trilateral [9] | 28.60 | 24.88 | 21.17 | 18.03 | 15.61 | 14.67 |
| | MNF [10] | 27.44 | 25.27 | 23.98 | **22.69** | **21.28** | **20.55** |
| | Ours | **30.28** | **26.48** | **24.04** | 21.71 | 19.55 | 18.54 |

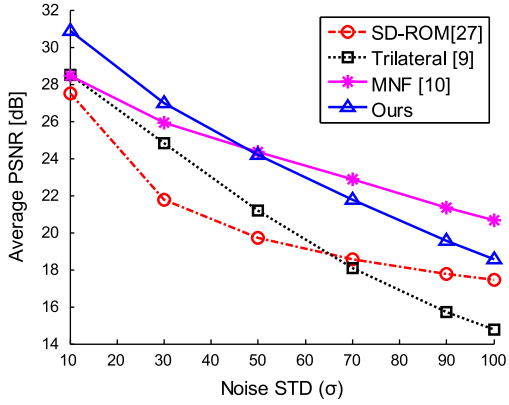


**Fig. 6.** Comparisons in terms of average PSNR corresponding to Table 5.

$$= \left| w_j \left( \lambda_1 \exp\left( -\frac{||X_i - Y_j||_2^2}{h_1^2} \right) \frac{2}{h_1^2} ||X_i - Y_j||_2 + \lambda_2 \left( 4/{h_2}^2 \left( 1 + \frac{||X_i - Y_j||_2^2}{h_2^2} \right)^3 \right) \left| \left| X_i - Y_j \right| \right|_2 \right) \right|$$

$$= w_j \left( \lambda_1 \exp\left( -\frac{||X_i - Y_j||_2^2}{h_1^2} \right) \frac{2}{h_2^2} + \lambda_2 \left( 4/h_2^2 \left( 1 + \frac{||X_i - Y_j||_2^2}{h_2^2} \right)^3 \right) \right) ||X_i - Y_j||_2 \quad \text{(A.6)}$$

Let $x$ denote $||X_i - Y_j||_2$ and take continuation strategy in Eq. (A.6), we get

$$|\phi_x| = w_j \left( \lambda_1 \exp\left( -\frac{x^2}{h_1^2} \right) \frac{2}{h_1^2} + \lambda_2 \left( 4/h_2^2 \left( 1 + \frac{x^2}{h_2^2} \right)^3 \right) \right) x \quad \text{(A.7)}$$

The function in (A.7) is continuous and derivable. Applying the L'Hospital's rule, we can get the following limitations for Eq. (A.7)

$$\lim_{x \to \infty} |\phi_x| = 0 \quad \text{(A.8)}$$

And their bounded maximum and minimum can be got by zeroing the derivative of the function in Eq. (A.7).

According to the above, the function in Eq. (A.6) is bounded. So the corresponding *IF* is bounded too. Therefore the gross error sensitivity defined in Eq. (A.4) is finite, which means that GNLMKIM is a robust estimator.

## References


[1] P. Chatterjee, P. Milanfar, Practical bounds on image denoising: from estimation to information, IEEE Trans. Image Process. 20 (2011) 1221–1233. (May).
[2] L.I. Rudin, et al., Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268. (November 1).
[3] C. Tomasi, R. Manduch, Bilateral filtering for gray and color images, in: International Conference on Computer Vision (ICCV), Bombay, India, 1998, pp. 839–846.
[4] A. Buades, et al., A non local algorithm for image denoising, in: International Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, pp. 60–65.
[5] A. Buades, et al., Image denoising methods. A new nonlocal principle, SIAM Rev.: Multiscale Model. Simul. 52 (2010) 113–147.
[6] K. Dabov, et al., Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (2007) 2080–2095.
[7] M. Aharon, et al., The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (2006) 4311–4322.
[8] P. Milanfar, A tour of modern image filtering: new insights and methods, both practical and theoretical, IEEE Signal Process. Mag. 30 (2013) 106–128.
[9] R. Garnett, et al., A universal noise removal algorithm with an impulse detector, IEEE Trans. Image Process. 14 (Nov 2005) 1747–1754.
[10] B. Li, et al., A new method for removing mixed noises, Sci. China Inf. Sci. 54 (2011) 51–59.
[11] P. Chatterjee, P. Milanfar, Is denoising dead? IEEE Trans. Image Process. 19 (2010) 895–911. (April).
[12] P.J. Huber, E.M. Ronchetti, Robust Statistics, 2 ed., John Wiley & Sons, Inc, New Jersey, 2009.
[13] K. Tan, et al., Robust image denoising using kernel-induced measures, in: The 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK, 2004, pp. 685–688.
[14] S. Sonnenburg, et al., Large scale multiple kernel learning, J. Mach. Learn. Res. 7 (Jul 2006) 1531–1565.
[15] A. Buades, et al., A review of image denoising algorithms, with a new one, SIAM Interdiscip. J.: Multiscale Model. Simul. 4 (2006) 490–530.
[16] H. Minh, et al., Mercer's Theorem, Feature Maps, and Smoothing Learning Theory, in: G. Lugosi, H. Simon (Eds.), Springer, Berlin/Heidelberg, 2006, pp. 154–168.
[17] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.
[18] B. Schölkopf, et al., Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.

[19] V. Roth, V. Steinhage, Nonlinear discriminant analysis using kernel functions, Adv. Neural Inf. Process. Syst. 12 (2000) 568–574.
[20] K.R. Muller, et al., An introduction to kernel-based learning algorithms, IEEE Trans. Neural Netw. 12 (2001) 181–201. (March).
[21] A. Ben Hamza, H. Krim, Image denoising: a nonlinear robust statistical approach, IEEE Trans. Signal Process. 49 (2001) 3045–3054.
[22] J. Sun, et al., Clustering with feature order preferences, Intell. Data Anal. 14 (2010) 479–495.
[23] V. Karnati, et al., Fast non-local algorithm for image denoising, 2009, in: 16th IEEE International Conference on Image Processing, vols. 1–6, 2009, pp. 3829–3832.
[24] R. Vignesh, et al., Fast non-local means (NLM) computation with probabilistic early termination, IEEE Signal Process. Lett. 17 (2010) 277–280. (March).
[25] N. Dowson, O. Salvado, Hashed non-local means for rapid image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 485–499.
[26] J.V. Manjon, et al., MRI denoising using non-local means, Med. Image Anal. 12 (2008) 514–523. (August).
[27] E. Abreu, S. K. Mitra, A signal-dependent rank ordered mean (Sd-Rom) filter – a new approach for removal of impulses from highly corrupted images, 1995, in: International Conference on Acoustics, Speech, and Signal Processing – Conference Proceedings, vols. 1–5, 1995, pp. 2371–2374.

**Zhonggui Sun** received his B.S. degree in mathematics from Shandong Normal University in 1996, and the Master Degree in Computer Software and Theory from Sichuan Normal University in 2006. Currently, he is an associate professor at Department of Mathematics Science at Liaocheng University. Also, he is currently a Ph.D. Student at the Department of Computer Science and Engineering at Nanjing university of Aeronautics & Astronautics (NUAA). His research interests include image processing, pattern recognition and machine learning.

**Songcan Chen** received the B.S. degree in mathematics from Hangzhou University (now merged into Zhejiang University) in 1983. In December 1985, he completed the M.Sc. degree in computer applications at Shanghai Jiaotong University and then worked at Nanjing University of Aeronautics and Astronautics (NUAA) in January 1986 as an assistant lecturer. There he received a Ph.D. degree in communication and information systems in 1997. Since 1998, as a full professor, he has been with the Department of Computer Science and Engineering at NUAA. His research interests include pattern recognition, image processing, machine learning and neural computing. In these fields, he has authored or coauthored over 150 scientific journal papers.

**Lishan Qiao** received the B.Sc. degree in mathematics from Liaocheng University (LU) in 2001. He received the M.Sc. degree in applied mathematics from Chengdu University of Technology in 2004, and then worked in LU to this day. In 2010, he received the Ph.D. degree in computer applications, at the Department of Computer Science & Engineering, Nanjing University of Aeronautics and Astronautics (NUAA). Currently, he is an associate professor at School of Mathematics Science, LU, and his research interests focus on image processing, pattern recognition and machine learning.