International Journal of Innovative
Computing, Information and Control
Volume **16**, Number **5**, October **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 1547–1558

IMAGE DENOISING USING LOW RANK MATRIX COMPLETION VIA BILINEAR GENERALIZED APPROXIMATE MESSAGE PASSING

Jingjing Si[1,3], Wenwen Sun[1] and Yinbo Cheng[2]

[1]School of Information Engineering
[3]Hebei Key Laboratory of Information Transmission and Signal Processing
Yanshan University
No. 438, West Hebei Avenue, Qinhuangdao 066004, P. R. China

[2]Ocean College
Hebei Agricultural University
No. 52, East Hebei Avenue, Qinhuangdao 066003, P. R. China
cyb@hebau.edu.cn

Received April 2020; revised August 2020

Abstract. *A robust denoising algorithm, which is capable of removing mixed noise from natural images, is proposed based on the bilinear generalized approximate message passing (BiGAMP). This algorithm utilizes the non-local self-similarities of natural images, changes the problem of removing mixed noise to a low-rank matrix completion problem, and takes use of BiGAMP to solve this inherently bilinear problem. Experimental results show that this BiGAMP-based image denoising algorithm is good at removing Gaussian noise mixed with impulsive noise. Its performance is better than the well-known block-matching and 3D filtering (BM3D) algorithm and some state-of-the-art image denoising algorithms via low-rank matrix completion.*

Keywords: Image denoising, Low rank matrix completion, Bilinear generalized approximate message passing, Mixed noise

1. **Introduction.** Image is the main way for people to obtain original information from the outside world. However, in the process of shooting, sampling, transmission and storage, image is often contaminated by external noise, resulting in degraded quality. Hence, image denoising plays an important role in the field of image processing.

Image noise encountered in the real world usually comes from multiple sources, which can be modeled as the mixture of Gaussian noise and impulse noise [1]. However, because these two types of noise have different characteristics, it is not easy to remove the mixed noise effectively. Most of existing image denoising methods focus on dealing with a special type of noise. For example, block-matching and 3D filtering (BM3D) [2] removes Gaussian noise with grouping and collaboratively filtering. Many local nonlinear operations, such as the adaptive median filter (AMF) [3], are developed specially for impulse noise removal.

A direct idea to remove the mixed noise is to use the two-step method, which estimates the impulse noise firstly and then discusses the removal of Gaussian noise. For example, in [4], median filter is used to detect the outliers, and then the K-SVD dictionary learning algorithm is used to solve the l_1-l_0 minimum optimization problem for Gaussian noise removal. Although the method can suppress the effect of mixed noise on image visual quality to some extent, it filters out many meaningful texture information. Facing this problem, some scholars devote to removing mixed noise without separating the sources [5].

DOI: 10.24507/ijicic.16.05.1547

Low-rank matrix completion (LRMC), which aims to recover the underlying low rank matrix from its degraded observation, has shown great application potential in the field of image processing, especially, in high-performance image restoration tasks [6-8]. It has been proved that the matrix formed by non-local similar patches in a natural image is with low rank [9,10]. In this way, noise removal from groups of matched image patches can be formulated as solving the problem of LRMC [11,12].

Bilinear generalized approximate message passing (BiGAMP), which is extended from the generalized approximate message passing (GAMP) [13], is a promising approach with impressive signal reconstruction results, and is applicable in some special cases of bilinear inference problem [14,15]. In this paper, we develop a robust image denoising algorithm, which is capable of removing mixed noise from the image, based on BiGAMP. This algorithm utilizes the non-local self-similarities of natural images, converts the problem of removing mixed noise to an LRMC problem, and then takes advantages of BiGAMP to solve this special bilinear problem.

The rest of the paper is organized as follows. Section 2 introduces the focused problem and the mathematical background. Section 3 presents the image denoising algorithm by LRMC via BiGAMP. Section 4 shows simulation results. Section 5 concludes this work.

2. **Mathematical Formulation.** Our focused problem can be expressed as to recover clean image $\boldsymbol{F}$ from noisy observation $\boldsymbol{I}$,

$$\boldsymbol{I} = \boldsymbol{F} + \boldsymbol{W}_G + \boldsymbol{W}_I \tag{1}$$

where $\boldsymbol{W}_G$ and $\boldsymbol{W}_I$ denote Gaussian noise and impulse noise. That is, our goal is to recover $\boldsymbol{F}$ by removing mixed noise $\boldsymbol{W}_G + \boldsymbol{W}_I$ from noisy image $\boldsymbol{I}$.

In LRMC, a few (possibly noise-corrupted) entries of a low-rank matrix are observed. The goal is to infer the missing entries from the observations. It can be formulated as to recover the low-rank matrix $\boldsymbol{Z}$ from its noisy observation matrix $\boldsymbol{Y}$,

$$\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{W} \tag{2}$$

where $\boldsymbol{W}$ is the sparse noise matrix.

BiGAMP is a novel algorithmic framework for the generalized bilinear inference problem. It estimates matrices $\boldsymbol{A} = [a_{mn}] \in \mathrm{R}^{M\times N}$ and $\boldsymbol{X} = [x_{nl}] \in \mathrm{R}^{N\times L}$ from a matrix observation $\boldsymbol{Y} \in \mathrm{R}^{M\times L}$ that is statistically coupled to their product $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$. Treat $\boldsymbol{A}$ and $\boldsymbol{X}$ as realizations of independent random matrices with known separable pdfs, i.e.,

$$p_A(\boldsymbol{A}) = \prod_m \prod_n p(a_{mn}) \tag{3}$$

$$p_X(\boldsymbol{X}) = \prod_n \prod_l p(x_{nl}) \tag{4}$$

and assume the likelihood function of $\boldsymbol{Z}$ is known and separable, i.e.,

$$p_{Y|Z}(\boldsymbol{Y}|\boldsymbol{Z}) = \prod_m \prod_l p(y_{ml}|z_{ml}) \tag{5}$$

The posterior distribution is

$$p_{A,X|Y}(\boldsymbol{A}, \boldsymbol{X}|\boldsymbol{Y}) \propto p_{Y|AX}(\boldsymbol{Y}|\boldsymbol{Z}) p_A(\boldsymbol{A}) p_X(\boldsymbol{X}) \tag{6}$$

where $\propto$ denotes equality up to a constant scale factor. Although the exact computation of $p_{A,X|Y}(\boldsymbol{A}, \boldsymbol{X}|\boldsymbol{Y})$ is generally prohibitive, BiGAMP can compute minimum mean-squared error (MMSE) estimates of both $\boldsymbol{A}$ and $\boldsymbol{X}$ [14].

In the following section, an image denoising algorithm by LRMC via BiGAMP is proposed. It converts the problem of removing mixed noise $\boldsymbol{W}_G + \boldsymbol{W}_I$ from noisy image $\boldsymbol{I}$ to a

series of LRMC problem of recovering $\left\{\hat{\boldsymbol{Z}}_i\right\}$ from noisy and incomplete observations $\{\boldsymbol{Y}_i\}$. Then, each noiseless complete low-rank matrix $\hat{\boldsymbol{Z}}_i = \hat{\boldsymbol{A}}_i\hat{\boldsymbol{X}}_i$ is recovered by estimating $\hat{\boldsymbol{A}}_i$ and $\hat{\boldsymbol{X}}_i$ from $\boldsymbol{Y}_i$ via BiGAMP.

3. **Image Denoising by LRMC via BiGAMP.** The process of our image denoising algorithm by LRMC via BiGAMP is shown in Algorithm 1. It includes three main steps, which will be described in detail in the following three subsections.

Algorithm 1. Image denoising by LRMC via BiGAMP

Inputs: Noisy image $\boldsymbol{I}$, patch size M, the number of non-local similar patches N_k, the initial window size w_{int}, the maximum window size $w_{\max}$, the maximum number of iterations $T_{\max}$, damping parameter β and stopping parameter ε.

Step 1: Patch grouping.

1) Apply Algorithm 2 on noisy image $\boldsymbol{I}$ to obtain pre-processed image $\hat{\boldsymbol{I}}$.

2) Partition $\hat{\boldsymbol{I}}$ into overlapping patches $\{\boldsymbol{q}_i\colon\ i = 1, \ldots, N_p\}$.

3) For each $\boldsymbol{q}_i$, find its similar patches group $\left\{\boldsymbol{q}_i^k\right\}_{k=1}^{N_k}$, and form corresponding low-rank matrix $\boldsymbol{Q}_i$.

Step 2: For each $\boldsymbol{Q}_i$, throw away unreliable pixels and get incomplete low-rank matrix $\boldsymbol{Y}_i$, $i = 1, \ldots, N_p$.

Step 3: Incomplete low-rank matrix completion via BiGAMP.

1) For each $\boldsymbol{Y}_i$, apply Algorithm 3 to estimate noiseless complete matrix $\hat{\boldsymbol{Z}}_i$, $i = 1, \ldots, N_p$.

2) Aggregate overlapped patches extracted and reshaped from $\left\{\hat{\boldsymbol{Z}}_i\colon\ i = 1, \ldots, N_p\right\}$ to form the recovered image $\hat{\boldsymbol{F}}$.

Outputs: The recovered image $\hat{\boldsymbol{F}}$.

3.1. **Patch grouping.** To avoid the presence of serious impulsive noise degrading the performance of patch matching, before block partition, AMF is taken as pre-filtering to remove obvious impulsive noise. Model the value of pixel (i, j) in image $\boldsymbol{I}$ as

$$\tilde{u}_{ij} = \begin{cases} d_{\min}, & \text{with probability } \gamma/2 \\ d_{\max}, & \text{with probability } \gamma/2 \\ u_{ij}, & \text{with probability } 1-\gamma \end{cases} \tag{7}$$

where $0 \leq \gamma \leq 1$ is the density level of the impulsive noise, $[d_{\min}, d_{\max}]$ is the dynamic range of image, and u_{ij} is the value of pixel (i, j) that is not stained by impulsive noise. Define a window W_{ij}^w of size $w \times w$ for pixel (i, j) as

$$W_{ij}^w = \{(a, b)\colon |a - i| \leq w \text{ and } |b - j| \leq w\} \tag{8}$$

The main process of AMF realized in our scheme is concluded as Algorithm 2. It identifies the pixel seriously corrupted by impulsive noise and replaces it by the median of pixels in its neighborhood.

Then, $\hat{\boldsymbol{I}}$ is partitioned into overlapping patches $\{\boldsymbol{q}_i\colon\ i = 1, \ldots, N_p\}$, and non-local similar patches group $\left\{\boldsymbol{q}_i^k\right\}_{k=1}^{N_k}$ is searched for each reference patch $\boldsymbol{q}_i$ by fast non-local block matching algorithm [16,17]. Firstly, the sum of squared difference (SSD) is calculated for

every patch $\boldsymbol{q}_j$ in the searching window of $\boldsymbol{q}_i$,

$$SSD(\boldsymbol{q}_i, \boldsymbol{q}_j) = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} [q_i(m,n) - q_j(m,n)]^2 \tag{9}$$

where $q_i(m,n)$ and $q_j(m,n)$ are the values of pixel (m,n) in patches $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$. Next, sort the values of SSD in ascending order. Take $\boldsymbol{q}_i$ as $\boldsymbol{q}_i^1$, and select patches that correspond to the first $N_k - 1$ smallest values of SSD as $\left\{\boldsymbol{q}_i^k\right\}_{k=2}^{N_k}$, to constitute the similar patches group $\left\{\boldsymbol{q}_i^k\right\}_{k=1}^{N_k}$ of $\boldsymbol{q}_i$. At last, reshape each $M \times M$ block $\boldsymbol{q}_i^k$ into $M^2 \times 1$ column vector $\bar{\boldsymbol{q}}_i^k$, and arrange $\bar{\boldsymbol{q}}_i^1, \bar{\boldsymbol{q}}_i^2, \ldots, \bar{\boldsymbol{q}}_i^{N_k}$ in juxtaposition to form matrix $\boldsymbol{Q}_i$,

$$\boldsymbol{Q}_i = \left[\bar{\boldsymbol{q}}_i^1, \bar{\boldsymbol{q}}_i^2, \ldots, \bar{\boldsymbol{q}}_i^{N_k}\right] \tag{10}$$

Since column vectors $\bar{\boldsymbol{q}}_i^1, \bar{\boldsymbol{q}}_i^2, \ldots, \bar{\boldsymbol{q}}_i^{N_k}$ have similar underlying structures, the rank of $\boldsymbol{Q}_i$ would be low and the variance of each row vector in $\boldsymbol{Q}_i$ would be very small. The denoising problem on image $\hat{\boldsymbol{I}}$ is now converted into the denoising problem on low-rank matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_{N_p}$.

Algorithm 2. AMF realized in our scheme

Inputs: Noisy image $\boldsymbol{I}$, the initial window size w_{int} and the maximum window size $w_{\max}$.
Initialization: set $w = w_{\text{int}}$.
For every pixel (i,j) in $\boldsymbol{I}$
  While $w \le w_{\max}$
    Compute the minimum value $d_{ij}^{\min,w}$, median value $d_{ij}^{\text{med},w}$ and maximum value $d_{ij}^{\max,w}$ of the pixels in window W_{ij}^w.
    If $d_{ij}^{\min,w} < d_{ij}^{\text{med},w} < d_{ij}^{\max,w}$
      If $d_{ij}^{\min,w} < \tilde{u}_{ij} < d_{ij}^{\max,w}$
        $\tilde{u}_{ij}$ keeps its original value
      Else
        Break
      End if
    Else
      $w = w + 2$
    End if
  End while
  Replace $\tilde{u}_{ij}$ with the median value $d_{ij}^{\text{med},w}$.
End for
Outputs: The pre-processed image $\hat{\boldsymbol{I}}$.

3.2. **Incomplete low-rank matrix construction.** In each matrix $\boldsymbol{Q}_i$, some pixels are unreliable since it may be damaged by impulsive noise, corrupted by Gaussian noise with large amplitude or from mismatched patches. Here, we set the values of these unreliable pixels to zeros, and discuss the restoration of low-rank matrix based on pixels with high reliability.

Pixel (u,v) in matrix $\boldsymbol{Q}_i$ is considered as reliable if its value $Q_i(u,v)$ is close to the mean $\bar{Q}_i(u)$ of the uth row. In detail, each element $Y_i(u,v)$ in resulting matrix $\boldsymbol{Y}_i$ is calculated as

$$Y_i(u,v) = \begin{cases} Q_i(u,v) & \text{if } \left|Q_i(u,v) - \bar{Q}_i(u)\right| \le \lambda \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where threshold λ is determined as

$$\lambda = \frac{1}{N_k} \sum_{v=1}^{N_k} \left| Q_i(u,v) - \bar{Q}_i(u) \right| \tag{12}$$

Since the matched image patches have similar underlying structures, each matrix $\boldsymbol{Y}_i$ would be a noisy version of a low-rank matrix with many missing elements.

With these steps, the problem of removing mixed noise from image $\boldsymbol{I}$ is now converted to the problem of recovering complete low-rank matrices $\left\{ \hat{\boldsymbol{Z}}_i : i = 1, \ldots, N_p \right\}$ from their noisy and incomplete observations $\{\boldsymbol{Y}_i : i = 1, \ldots, N_p\}$.

3.3. **Incomplete low-rank matrix completion with BiGAMP.** Here, we seek to recover a rank-N_i matrix $\hat{\boldsymbol{Z}}_i \in \mathrm{R}^{M_i \times L_i}$ from noisy incomplete matrix $\boldsymbol{Y}_i$, where $N_i \ll \min(M_i, L_i)$. Model the complete matrix $\boldsymbol{Z}_i$ as the product of matrices $\boldsymbol{A}_i \in \mathrm{R}^{M_i \times N_i}$ and $\boldsymbol{X}_i \in \mathrm{R}^{N_i \times L_i}$ with priors of the decoupled form in (3) and (4), where $\boldsymbol{Z}_i$ is probabilistically related to matrix $\boldsymbol{Y}_i$ through a likelihood $p(\boldsymbol{Y}_i | \boldsymbol{Z}_i)$ of the decoupled form in (5). We take BiGAMP to infer $\boldsymbol{A}_i$ and $\boldsymbol{X}_i$ from $\boldsymbol{Y}_i$. The product of estimates $\hat{\boldsymbol{A}}_i$ and $\hat{\boldsymbol{X}}_i$ yields estimate $\hat{\boldsymbol{Z}}_i$ of the noiseless complete matrix $\boldsymbol{Z}_i$.

The BiGAMP algorithm with scalar variances realized in our scheme is concluded as Algorithm 3. Instead of storing and processing a number of element-wise variance terms, Algorithm 3 uses uniform scalar variances across the elements to reduce the memory and complexity. It uses the maximum number of iterations and the normalized change in the residual as stopping conditions, and incorporates damping to guarantee convergence.

We model the entries of matrix $\boldsymbol{A}_i$ using i.i.d. standard normal Gaussian prior and the entries of matrix $\boldsymbol{X}_i$ using i.i.d. zero-mean Gaussian prior, i.e.,

$$\forall m, n \quad p(a_{mn}) = N(0,1) \tag{13}$$

$$\forall n, l \quad p(x_{nl}) = N\left(0, v_0^x\right) \tag{14}$$

Moreover, we model the likelihood function as

$$p(y_{ml}|z_{ml}) = \begin{cases} N\left(z_{ml}, v^w\right) & (m,l) \in \Omega_i \\ 1_{y_{ml}} & (m,l) \notin \Omega_i \end{cases} \tag{15}$$

where Ω_i is the set of indices of the remained elements in matrix $\boldsymbol{Y}_i$ and 1_y denotes a point mass at $y = 0$. By looking up Table 1 in [13], the scalar estimation functions $g_{out,ml}(\cdot)$, $g_{in,a,mn}(\cdot)$, $g_{in,x,nl}(\cdot)$ and their derivatives utilized in Algorithm 3 are set as

$$g_{out,ml}\left(y_{ml}, p_{ml}(t); v^p(t)\right) = \begin{cases} \dfrac{y_{ml} - p_{ml}(t)}{v^p(t) + v^w} & (m,l) \in \Omega_i \\ 0 & (m,l) \notin \Omega_i \end{cases} \tag{16}$$

$$-g'_{out,ml}\left(y_{ml}, p_{ml}(t); v^p(t)\right) = \begin{cases} \dfrac{1}{v^p(t) + v^w} & (m,l) \in \Omega_i \\ 0 & (m,l) \notin \Omega_i \end{cases} \tag{17}$$

$$g_{in,a,mn}\left(q_{mn}(t); v^q(t)\right) = \frac{q_{mn}(t)}{v^q(t) + 1} \tag{18}$$

$$v^q(t) g'_{in,a,mn}\left(q_{mn}(t); v^q(t)\right) = \frac{v^q(t)}{v^q(t) + 1} \tag{19}$$

$$g_{in,x,nl}\left(r_{nl}(t); v^r(t)\right) = \frac{r_{nl}(t) \cdot v_0^x}{v^r(t) + v_0^x} \tag{20}$$

$$v^r(t) g'_{in,x,nl}\left(r_{nl}(t); v^r(t)\right) = \frac{v^r(t) \cdot v_0^x}{v^r(t) + v_0^x} \tag{21}$$

v_0^x and v^w are tuned using the expectation-maximization (EM) approach as in [18].

The steps in Algorithm 3 can be interpreted as follows. (R1)-(R2) compute an estimate $\boldsymbol{P}(t)$ of $\boldsymbol{Z}$ and corresponding variance $v^p(t)$. (R3) computes $\boldsymbol{U}(t)$ that will be used in (R17). (R4)-(R5) compute scaled residual $\boldsymbol{s}(t)$ and variance $v^s(t)$. (R6)-(R8) compute $\boldsymbol{r}(t)$ and variance $v^r(t)$, where $\boldsymbol{r}(t)$ can be interpreted as a $v^r(t)$-variance-AWGN corrupted observation of $\boldsymbol{X}$. (R9)-(R11) compute $\boldsymbol{q}(t)$ and variance $v^q(t)$, where $\boldsymbol{q}(t)$ can be interpreted as a $v^q(t)$-variance-AWGN corrupted observation of $\boldsymbol{A}$. (R12)-(R13) estimate $\boldsymbol{X}(t)$ and variance $v^x(t)$ from $\boldsymbol{r}(t)$ and $v^r(t)$. (R14)-(R15) estimate $\boldsymbol{A}(t)$ and variance

Algorithm 3. BiGAMP algorithm with scalar variances realized in our scheme

Inputs: observation matrix $\boldsymbol{Y} = \boldsymbol{Y}_i$, the maximum number of iterations $T_{\max}$, damping parameter β and stopping parameter ε.

Initialization: $v^p(0) = 0$, $v^s(0) = 0$; $\forall m, n, l$ $s_{ml}(0) = 0$, $\bar{x}_{nl}(0) = 0$, $\bar{a}_{mn}(0) = 0$; choose $v^x(1)$, $v^a(1)$; $\forall m, n, l$ choose $a_{mn}(1)$, $x_{nl}(1)$.

For $t = 1, 2, \ldots, T_{\max}$

1) Output linear step:

$$v^p(t) = \left(\frac{\|\boldsymbol{A}(t)\|_F^2}{M_i} v^x(t) + \frac{\|\boldsymbol{X}(t)\|_F^2}{L_i} v^a(t) + N \cdot v^a(t) \cdot v^x(t)\right) \cdot \beta + v^p(t-1)(1-\beta) \tag{R1}$$

$$\forall m, l \; p_{ml}(t) = \sum_{n=1}^{N_i} a_{mn}(t) x_{nl}(t) - v^p(t) s_{ml}(t-1) \tag{R2}$$

$$u_{ml}(t) = y_{ml} - \sum_{n=1}^{N_i} a_{mn}(t) x_{nl}(t) \tag{R3}$$

2) Output nonlinear step:

$$\forall m, l \; s_{ml}(t) = g_{out,ml}\left(y_{ml}, p_{ml}(t); v^p(t)\right) \cdot \beta + s_{ml}(t-1) \cdot (1-\beta) \tag{R4}$$

$$v^s(t) = \left(-\frac{1}{M_i L_i} \sum_{m=1}^{M_i} \sum_{l=1}^{L_i} g'_{out,ml}\left(y_{ml}, p_{ml}(t); v^p(t)\right)\right) \cdot \beta + v^s(t-1) \cdot (1-\beta) \tag{R5}$$

3) Input linear step:

$$v^r(t) = \frac{N_i}{v^s(t) \cdot \|\boldsymbol{A}(t)\|_F^2} \tag{R6}$$

$$\forall n, l \; \bar{x}_{nl}(t) = x_{nl}(t) \cdot \beta + \bar{x}_{nl}(t-1) \cdot (1-\beta) \tag{R7}$$

$$r_{nl}(t) = \bar{x}_{nl}(t) \cdot \left(1 - \frac{M_i N_i \cdot v^a(t)}{\|\boldsymbol{A}(t)\|_F^2}\right) + v^r(t) \cdot \sum_{m=1}^{M_i} a_{mn}(t) s_{ml}(t) \tag{R8}$$

$$v^q(t) = \frac{M_i}{v^s(t) \cdot \|\boldsymbol{X}(t)\|_F^2} \tag{R9}$$

$$\forall m, n \; \bar{a}_{mn}(t) = a_{mn}(t) \cdot \beta + \bar{a}_{mn}(t-1) \cdot (1-\beta) \tag{R10}$$

$$q_{mn}(t) = \bar{a}_{mn}(t) \cdot \left(1 - \frac{N_i L_i \cdot v^x(t)}{\|\boldsymbol{X}(t)\|_F^2}\right) + v^q(t) \cdot \sum_{l=1}^{L_i} x_{nl}(t) s_{ml}(t) \tag{R11}$$

4) Input nonlinear step:

$$\forall n, l \; x_{nl}(t+1) = g_{in,x,nl}\left(r_{nl}(t); v^r(t)\right) \tag{R12}$$

$$v^x(t) = \frac{1}{N_i L_i} v^r(t) \sum_{n=1}^{N_i} \sum_{l=1}^{L_i} g'_{in,x,nl}\left(r_{nl}(t); v^r(t)\right) \tag{R13}$$

$$\forall m, n \; a_{mn}(t+1) = g_{in,a,mn}\left(q_{mn}(t); v^q(t)\right) \tag{R14}$$

$$v^a(t) = \frac{1}{M_i N_i} v^q(t) \sum_{m=1}^{M_i} \sum_{n=1}^{N_i} g'_{in,a,mn}\left(q_{mn}(t); v^q(t)\right) \tag{R15}$$

$$\forall m, l \; u_{ml}(t+1) = y_{ml} - \sum_{n=1}^{N_i} a_{mn}(t) x_{nl}(t) \tag{R16}$$

If $\|\boldsymbol{U}(t+1) - \boldsymbol{U}(t)\|_F^2 / \|\boldsymbol{U}(t+1)\|_F^2 < \varepsilon$ break (R17)

End for

Outputs: $\hat{\boldsymbol{A}}_i = \boldsymbol{A}(t+1)$, $\hat{\boldsymbol{X}}_i = \boldsymbol{X}(t+1)$.

$v^a(t)$ from $\boldsymbol{q}(t)$ and $v^q(t)$. (R16) updates $\boldsymbol{U}(t+1)$ and (R17) controls the early stop of iteration based on the normalized change of residual $\boldsymbol{U} = \boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}$.

After recovering $\left\{\hat{\boldsymbol{Z}}_i : i = 1, \ldots, N_p\right\}$ with Algorithm 3 from $\{\boldsymbol{Y}_i : i = 1, \ldots, N_p\}$, the overlapping patches are reshaped and the denoised image $\hat{\boldsymbol{F}}$ is synthesized. Since a pixel in the recovered image might be covered by several denoised patches, the value is determined by taking the average of the denoised patches at this pixel. It helps suppress the possible artifacts in the neighborhood of the boundaries of patches.

4. **Simulation Results.** We compare our proposed image denoising algorithm (referred to as LRMC-BiGAMP) with three state-of-the-art denoising algorithms. These three algorithms are modified BM3D (MBM3D) which pre-processes the noisy image with AMF before denoising it with BM3D for the sake of fair comparison, matrix completion via BiGAMP (MC-BiGAMP) [15] and low-rank matrix completion via alternating direction method of multipliers (LRMC-ADMM) [19]. All implementations are coded in Matlab R2016a. Computations are performed on Windows 10 operating system with Intel(R) Core(TM) i5-8400 CPU at 2.8 GHz and 8 GB memory. Standard grayscale images "Lena", "Airplane", "Boat", "Barbara", "House" and "Peppers" of 512×512 pixels are taken as the testing images. Zero-mean additive white Gaussian noise with variance σ^2 and impulsive noise of density γ are added to generate the observed noisy images.

4.1. **Parameters setting.** First, we discuss the proper choice of the initial window size w_{int} and the maximum window size $w_{\max}$ in Algorithm 2. Considering the symmetry, we set w_{int} as 3, and discuss the choice of $w_{\max}$ with experiments. Taking "Peppers" as testing image, Figure 1 shows PSNR achieved by LRMC-BiGAMP versus $w_{\max}$ under different γ. As shown, under each γ, PSNR maintains steadily after $w_{\max} = 25$. Hence, $w_{\max}$ is set as 25 in the following experiments.

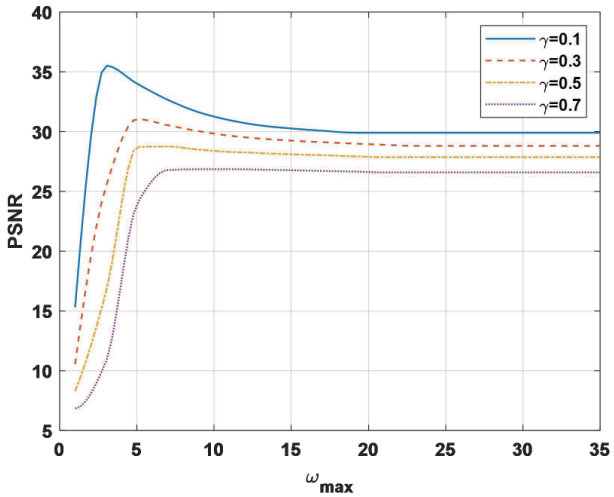


FIGURE 1. PSNR achieved by LRMC-BiGAMP versus $w_{\max}$ under different γ

Next, we consider the setting of parameters in block matching. Experiments show that, at low noise levels, superior performance can be achieved with small patch size M and small number of non-local similar blocks N_k; whereas, at high noise levels, large M and large N_k are appropriate. With these experiences, we set M as 6, 7, 8, 9, and N_k as 70, 90, 120, 140, when $\sigma^2 \leq 20$, $20 < \sigma^2 \leq 40$, $40 < \sigma^2 \leq 60$, $\sigma^2 > 60$ respectively.

Finally, we set $T_{\max} = 200$, $\beta = 0.8$ and $\varepsilon = 10^{-4}$ in Algorithm 3 according to the experimental experiences in the study of BiGAMP.

4.2. **Comparative experiments.** Firstly we compare the objective denoising performance of LRMC-BiGAMP with MBM3D, MC-BiGAMP, and LRMC-ADMM. The quality of the recovered image is evaluated by peak signal to noise ratio (PSNR) and structural similarity index measurement (SSIM). The values are averaged over 10 executions under the same situation. Tables 1 and 2 show the PSNR and SSIM achieved by these four algorithms at $\sigma^2 = 10$ and 30 when $\gamma = 0.1$. Tables 3 and 4 show the PSNR and SSIM

TABLE 1. PSNR (dB) and SSIM achieved when $\sigma^2 = 10$ and $\gamma = 0.1$

Image	MBM3D		MC-BiGAMP		LRMC-ADMM		LRMC-BiGAMP	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	24.07	0.7704	29.10	0.8780	26.83	0.7822	**31.75**	**0.9401**
Airplane	25.60	0.8572	26.36	0.8697	26.33	0.7796	**30.57**	**0.9399**
Boat	25.14	0.8203	26.32	0.8387	25.68	0.7653	**28.79**	**0.9136**
Barbara	22.86	0.7956	25.08	0.8193	23.20	0.7227	**27.46**	**0.9204**
House	24.03	0.6520	30.95	0.7841	25.51	0.6696	**32.14**	**0.8628**
Peppers	23.89	0.7654	27.27	0.7840	26.95	0.7983	**31.30**	**0.9453**

TABLE 2. PSNR (dB) and SSIM achieved when $\sigma^2 = 30$ and $\gamma = 0.1$

Image	MBM3D		MC-BiGAMP		LRMC-ADMM		LRMC-BiGAMP	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	21.44	0.4778	25.96	0.7642	24.74	0.6479	**28.74**	**0.8502**
Airplane	20.79	0.5262	24.04	0.7660	24.54	0.6616	**27.50**	**0.8530**
Boat	20.44	0.5585	23.96	0.6680	23.96	0.6680	**26.47**	**0.8065**
Barbara	21.72	0.5234	22.49	0.6611	22.16	0.6358	**25.02**	**0.8241**
House	21.28	0.4832	25.46	0.5425	24.52	0.6059	**28.86**	**0.7771**
Peppers	21.45	0.5128	23.44	0.5561	24.83	0.6698	**28.17**	**0.8620**

TABLE 3. PSNR (dB) and SSIM achieved when $\sigma^2 = 10$ and $\gamma = 0.3$

Image	MBM3D		MC-BiGAMP		LRMC-ADMM		LRMC-BiGAMP	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	22.52	0.7404	27.43	0.8500	26.64	0.7718	**31.12**	**0.9280**
Airplane	23.12	0.8185	24.96	0.8457	25.98	0.7722	**29.76**	**0.9279**
Boat	23.49	0.8083	24.86	0.7993	25.38	0.7558	**28.16**	**0.9006**
Barbara	21.45	0.7356	24.40	0.7928	22.91	0.7078	**26.36**	**0.8964**
House	21.89	0.6564	29.81	0.7510	25.08	0.6000	**31.54**	**0.8528**
Peppers	21.83	0.7452	25.33	0.7462	26.64	0.7871	**30.41**	**0.9354**

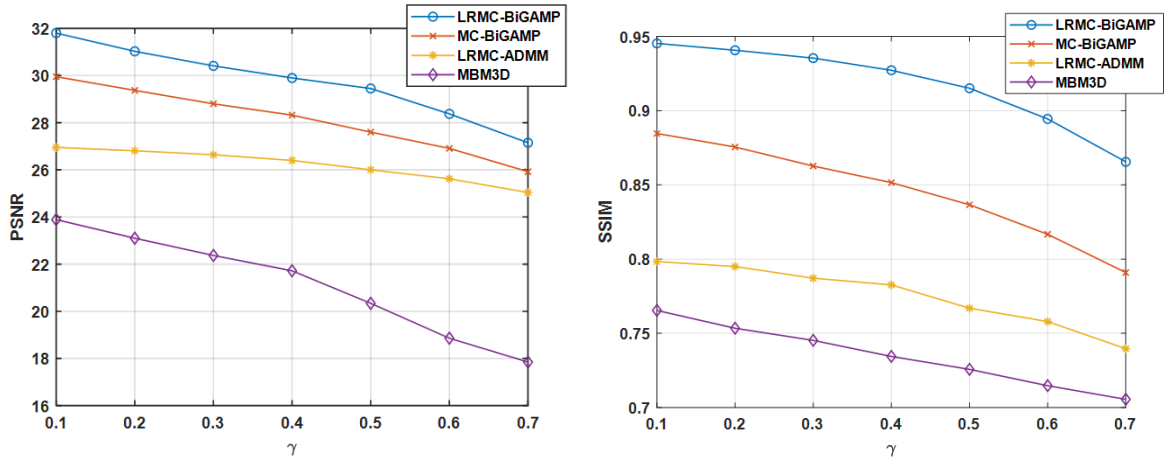
TABLE 4. PSNR (dB) and SSIM achieved when $\sigma^2 = 30$ and $\gamma = 0.3$

Image	MBM3D		MC-BiGAMP		LRMC-ADMM		LRMC-BiGAMP	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	20.07	0.4180	25.38	0.7436	24.15	0.6129	**28.07**	**0.8165**
Airplane	19.30	0.4186	23.07	0.7401	23.89	0.6213	**26.78**	**0.8218**
Boat	18.94	0.4923	23.14	0.6994	23.38	0.6234	**25.67**	**0.7762**
Barbara	17.37	0.4057	22.08	0.6313	21.74	0.6008	**24.42**	**0.7638**
House	19.40	0.4024	24.69	0.5080	24.23	0.5799	**27.94**	**0.7409**
Peppers	19.04	0.4153	22.62	0.5215	24.20	0.6411	**27.28**	**0.8288**

achieved by these four algorithms at $\sigma^2 = 10$ and 30 when $\gamma = 0.3$. As shown in Tables 1-4, for all testing images and under each noise level, both PSNR and SSIM achieved by LRMC-BiGAMP are much higher than those achieved by the other three algorithms.

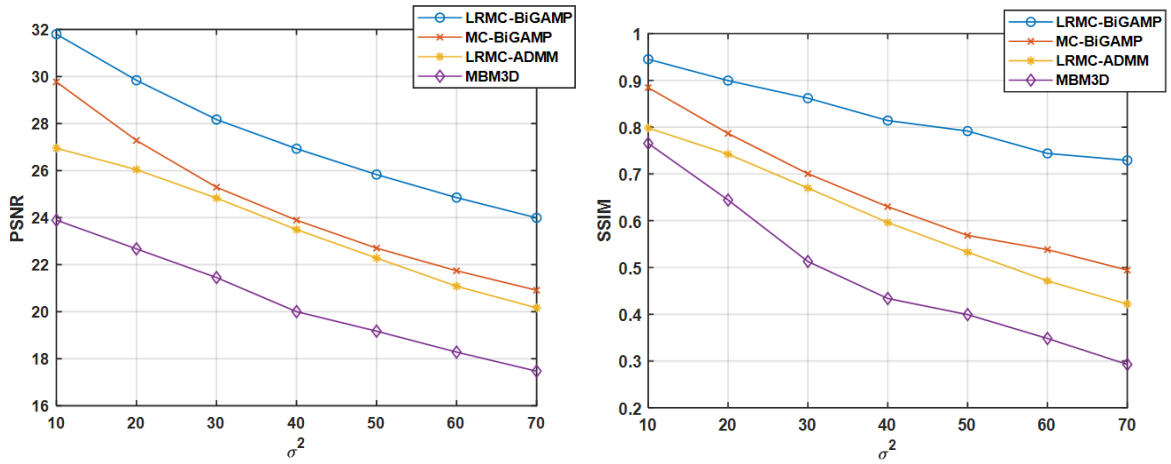
Next, the robustness of LRMC-BiGAMP to impulse noise and Gaussian noise is analyzed. Taking "Peppers" as testing image, Figure 2 shows the PSNR and SSIM achieved by LRMC-BiGAMP, MBM3D, MC-BiGAMP, and LRMC-ADMM under fixed $\sigma^2 = 10$ and different γ. Figure 3 shows the PSNR and SSIM achieved by these four algorithms under fixed $\gamma = 0.1$ and different σ^2. As shown, under each noise level, LRMC-BiGAMP achieves higher PSNR and SSIM than the other three algorithms. When σ^2 is changed, the variations of PSNR and SSIM are larger than those caused by the change of γ. That is, LRMC-BiGAMP is more robust to impulse noise but a bit sensitive to Gaussian noise.

Finally, we compare the visual quality of the images recovered by MBM3D, MC-BiGAMP, LRMC-ADMM and LRMC-BiGAMP. Taking "Airplane" as testing image, Figure 4 and Figure 5 show images recovered by these four algorithms at $\gamma = 0.1$ when σ^2 is 10 and 30. Figure 6 and Figure 7 show images recovered by these four algorithms at



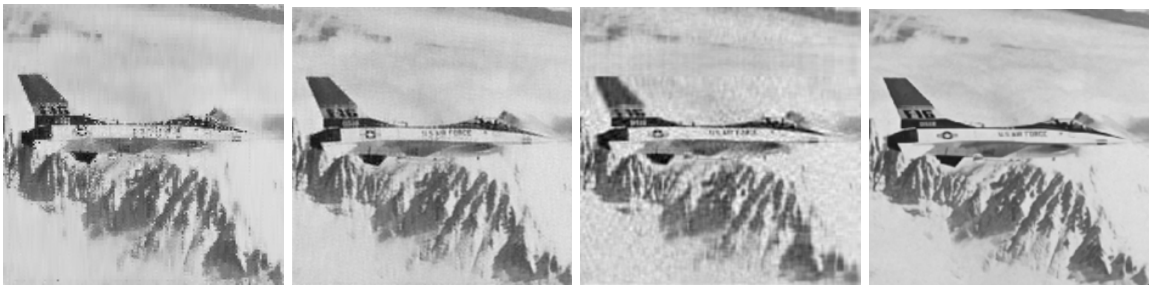
(a) PSNR vs γ (b) SSIM vs γ

FIGURE 2. PSNR and SSIM achieved under different γ when $\sigma^2 = 10$



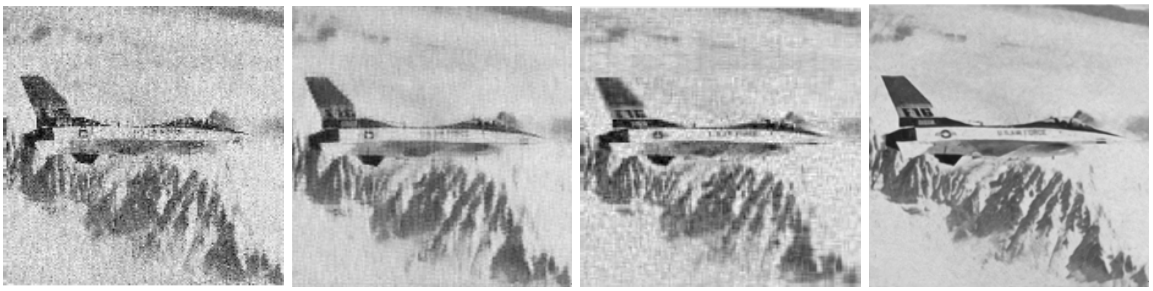
(a) PSNR vs σ^2 (b) SSIM vs σ^2

FIGURE 3. PSNR and SSIM achieved under different σ^2 when $\gamma = 0.1$



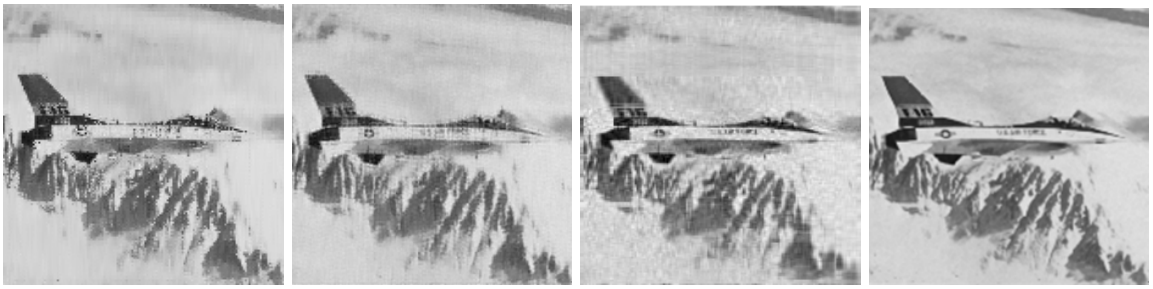
(a) MBM3D (b) MC-BiGAMP (c) LRMC-ADMM (d) LRMC-BiGAMP

FIGURE 4. Recovered images when $\sigma^2 = 10$ and $\gamma = 0.1$



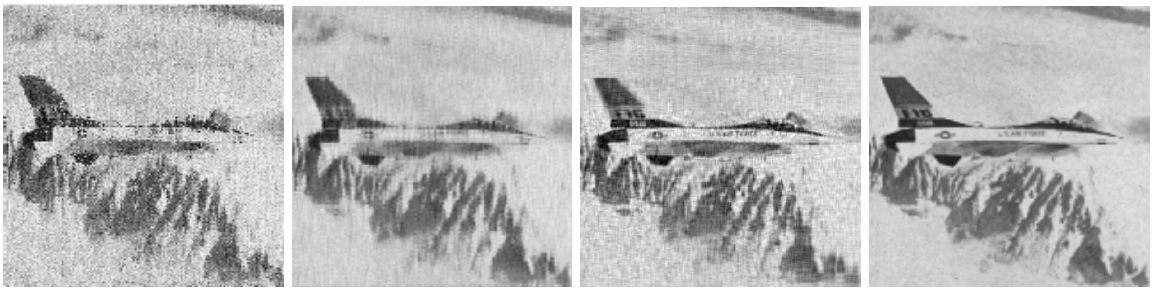
(a) MBM3D (b) MC-BiGAMP (c) LRMC-ADMM (d) LRMC-BiGAMP

FIGURE 5. Recovered images when $\sigma^2 = 30$ and $\gamma = 0.1$



(a) MBM3D (b) MC-BiGAMP (c) LRMC-ADMM (d) LRMC-BiGAMP

FIGURE 6. Recovered images when $\sigma^2 = 10$ and $\gamma = 0.3$



(a) MBM3D (b) MC-BiGAMP (c) LRMC-ADMM (d) LRMC-BiGAMP

FIGURE 7. Recovered images when $\sigma^2 = 30$ and $\gamma = 0.3$

$\gamma = 0.3$ when σ^2 is 10 and 30. As shown, images recovered by LRMC-BiGAMP have higher visual quality than those reconstructed by the other three algorithms. In detail, different from the state-of-the-art denoiser BM3D, LRMC-ADMM suppresses the mixed noise well. Compared to MC-BiGAMP, LRMC-BiGAMP retains more detailed feature information in the reconstructed images by taking use of the self-similarity in the image. Compared to LRMC-ADMM, LRMC-BiGAMP shows a more stable performance for mixed noise removal due to the perfect reconstruction ability of BiGAMP.

5. **Conclusions.** We propose a robust image denoising algorithm based on BiGAMP to remove mixed noise. This algorithm utilizes the non-local self-similarities of images, converts the problem of removing mixed noise to a low-rank matrix completion problem, and takes advantages of BiGAMP to solve this special bilinear problem. The effectiveness of our proposed BiGAMP-based denoising algorithm on removing Gaussian noise mixed with impulsive noise is validated with simulation experiments. It compares favorably against BM3D and some existing image denoising algorithms via LRMC. Meantime, it can preserve more detailed information in the recovered image.

Since parameters utilized in BiGAMP are tuned with EM approach, the runtime of our proposed BiGAMP-based image denoising algorithm is high. Computationally efficient image denoising algorithm based on Bayesian estimation is one subject of our future work.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (61701429) and the Natural Science Foundation of Hebei Province (F2018203137).

REFERENCES

[1] G. Shen, Z. Han and Y. Tang, Robust video denoising by low-rank decomposition and modeling noises with mixture of Gaussian, *Proc. of IEEE International Conference on Robotics and Biomimetics*, Bali, Indonesia, pp.2226-2231, 2014.

[2] K. Dabov, V. Foi, A. Katkovnik et al., Image denoising by sparse 3D transform-domain collaborative filtering, *IEEE Trans. Image Processing*, vol.16, no.8, pp.2080-2095, 2007.

[3] S. Zhang, X. Li and C. Zhang, Modified adaptive median filtering, *Proc. of International Conference on Intelligent Transportation, Big Data and Smart City*, Xiamen, China, 2018.

[4] Y. Xiao, T. Zeng, J. Yu et al., Restoration of images corrupted by mixed Gaussian-impulse noise via l_1-l_0 minimization, *Pattern Recognition*, vol.44, no.8, pp.1708-1720, 2011.

[5] J. Liu, X. Tai, H. Huang et al., A weighted dictionary learning model for denoising images corrupted by mixed noise, *IEEE Trans. Image Processing*, vol.22, no.3, pp.1108-1120, 2012.

[6] F. Nie, Z. Hu and X. Li, Matrix completion based on non-convex low rank approximation, *IEEE Trans. Image Processing*, vol.28, no.5, pp.2378-2388, 2018.

[7] Y. Hu, X. Liu and M. Jacob, A generalized structured low-rank matrix completion algorithm for MR image recovery, *IEEE Trans. Medical Imaging*, vol.38, no.8, pp.1841-1851, 2019.

[8] E. Hu and J. Kwok, Low-rank matrix learning using biconvex surrogate minimization, *IEEE Trans. Neural Networks and Learning Systems*, vol.30, no.11, pp.3517-3527, 2019.

[9] M. Nejati, S. Samavi, H. Derksen et al., Denoising by low-rank and sparse representations, *Journal of Visual Communication and Image Representation*, vol.36, pp.28-39, 2016.

[10] S. Huang and H. Wolkowicz, Low-rank matrix completion using nuclear norm minimization and facial reduction, *Journal of Global Optimization*, vol.72, pp.5-26, 2018.

[11] H. Zhang, J. Qian, B. Zhang et al., Low-rank matrix recovery via modified Schatten-p norm minimization with convergence guarantees, *IEEE Trans. Image Processing*, vol.29, pp.3131-3142, 2020.

[12] L. Fan, R. Meng, Q. Guo et al., Image denoising by low-rank approximation with estimation of noise energy distribution in SVD domain, *IET Image Processing*, vol.13, no.4, pp.680-691, 2019.

[13] S. Rangan. Generalized approximate message passing for estimation with random linear mixing, *Proc. of IEEE International Symposium on Information Theory*, St. Petersburg, USA, pp.2168-2172, 2011.

[14] J. Paker, P. Schniter and V. Cevher, Bilinear generalized approximate message passing – Part I: Derivation, *IEEE Trans. Signal Processing*, vol.62, no.22, pp.5839-5853, 2014.

[15] J. T. Paker, P. Schniter and V. Cevher, Bilinear generalized approximate message passing – Part II: Applications, *IEEE Trans. Signal Processing*, vol.62, no.22, pp.5854-5867, 2014.
[16] A. Parekh and I. Selesnick, Enhanced low-rank matrix approximation, *IEEE Signal Processing Letters*, vol.23, no.4, pp.493-497, 2016.
[17] T. Hiraoka, H. Nonaka and Y. Tsurunari, A high-speed method for generating labyrinth images using smoothing filters with different window sizes, *ICIC Express Letters*, vol.13, no.8, pp.711-717, 2019.
[18] J. Vila and P. Schniter, Expectation-maximization Gaussian-mixture approximate message passing, *IEEE Trans. Signal Processing*, vol.61, no.19, pp.4658-4672, 2013.
[19] Q. Yao and J. Kwok, Colorization by patch-based local low-rank matrix completion, *Proc. of the 29th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, Austin, TX, USA, pp.1959-1965, 2015.