Block Compressive Sensing for Solder Joint Images with Wavelet Packet Thresholding

Hui-Huang Zhao, Paul L. Rosin, and Yu-Kun Lai

Abstract—This paper provides a novel method which can achieve better results in solder joint imagery compression and reconstruction. Wavelet packet decomposition is used to generate some frequency coefficients of images. The higher and lower frequency coefficients of the reconstruction signal are used separately to improve the reconstruction performance. A threshold which only relates to the higher frequency coefficients is defined to remove the noise in the reconstruction result in each iteration. A new control factor is further defined to control the threshold value. The control factor relates to the wavelet packet low frequency coefficients, and is updated by the wavelet packet low frequency coefficients in each iteration. Experimental results reveal that the proposed algorithm is able to improve performance in terms of peak signal to noise ratio (PSNR) and structural similarity (SSIM), compared to classical algorithms in reconstruction of different types of solder joint images. When the sample rate is increased the proposed method improves reconstruction results and maintains low computational cost. The proposed algorithm can retain more image structure and achieve better results than some common methods.

Index Terms—Solder Joint Image, Block Compressive Sensing (CS), Orthogonal Matching Pursuit, Greedy Basis Pursuit, Subspace Pursuit, Compressive Sampling Patching Pursuit, Wavelet Packet Thresholding

I. INTRODUCTION

Surface mount technology (SMT) components are a key part of electronic products. Their assembly quality greatly affects the quality of the products. To improve the inspection rate of solder joint defects (such as pseudo-solder, insufficient solder), image compression, image segmentation [26], image enhancement and image filtering, etc. are used in automatic optical inspection (AOI) [1], [31]. Compressive Sensing (CS) is a sampling paradigm that provides signal compression at a significantly lower rate than the Nyquist rate [9], [10]. It has been successfully applied in a wide variety of applications in recent years, including image processing [5], Synthetic Aperture Radar (SAR) [3], Magnetic Resonance Imaging (MRI) [21], video processing [34], color images [2], polynomial expansion [23], information security [33] and solder joint image compression [36]. In [22], the authors proposed an adaptive observation matrix for sparse sampling of ultrasonic wave signals which were analyzed in phased array structural health monitoring. The authors in [17] proposed a novel reconstruction method for X-rays based on CS. [35] proposed a solder joint image compression method and used different square block dimensions (4, 8 or 16) when the image size is 256 × 256.

The success of deep convolutional neural networks (DCNNs) in computer vision has also raised interest in Compressive Sensing. [27], [29] both proposed a deep learning approach for accelerating MRI using a large number of existing high quality MR images as the training datasets. [18] proposed a novel DCNN CS method. In their method, the DCNN is designed to learn to take measurements and recover signals. [30] developed a novel CS method based on the Deep Residual Reconstruction Network (DR²-Net). DR²-Net uses two observations to reconstruct the image from its CS measurement. Those methods based on deep CNNs need a large number of existing images and much time to train the model. However, the number of sample defect images is usually very limited, so it can be impractical to apply this approach to solder joint image compressive sensing.

In order to improve the performance in image compressive sensing, [13] proposed and studied block compressive sensing for natural images and this method involves Wiener filtering and projection onto the convex set and hard thresholding in the transform domain. For 512 × 512 size images, the author suggested block dimension 32 and proposed a BPL (Block Projected Landweber) method with a variant of projected Landweber (PL) iteration and smoothing [19], [4], [32] and [16] studied the block compressed sensing with projected Landweber (PL). [24] proposed a block compressed sensing method based on iterative re-weighted l1 norm minimization. During those methods the row and column dimensions of the measurement matrix size are the square of the block size. Thus the approach requires substantially more memory as the block size increases.

In this paper, we develop a novel CS algorithm named BCS_WP_SPL. The three main contributions of this paper are summarized as follows:

- Wavelet packet decomposition is used to generate some frequency coefficients of signals. We separately use its higher and lower frequency coefficients of the reconstruction signal to improve the reconstruction performance.
- We define a threshold which only relates to the higher frequency coefficients to remove the noise in the reconstruction result in each iteration.
- We define a new control factor which is used to control the threshold value. The control factor relates to the wavelet packet’s low frequency coefficients which are used to update it in each iteration.

The rest of this paper is organized as follows. In section II, we introduce related work on CS. In section III, we describe the
BCS_WPL method for image compression. Experimental results and comparison are shown in section IV. Finally, we conclude our paper in section V.

II. RELATED WORK

The major challenge in CS is to approximate a signal given a vector of samples. Given a signal \( x \in \mathbb{R}^{N \times N} \), we want to recover \( x \) from \( y = \Phi x \), where \( \Phi \in \mathbb{R}^{M \times N} \) \((M < N)\) is a measurement matrix. If \( x \) is sufficiently sparse, \( x \) can be exactly recovered with CS theory. Otherwise, \( x \) can be made sparse by applying orthogonal transforms, for example, the Discrete Cosine Transform (DCT), from \( \hat{x} = \Psi x \), where \( \Psi \in \mathbb{R}^{N \times N} \) is an orthogonal basis matrix. Recovery of \( x \) with the smallest \( l_0 \) norm consistent with the observed \( y \) is an NP-complete problem. Usually, \( x \) can be recovered with an \( l_1 \) optimization:

\[
\begin{aligned}
\text{minimize} & \| \hat{x} \|_1 \\
\text{subject to:} & \ y = \Phi \Psi^{-1} \hat{x}
\end{aligned}
\]  

(1)

There are many methods available for solving the problem in Eq. 1. One common method is based on a projection which forms \( \hat{x} \) by successive projection and thresholding. Given an initial approximation \( \hat{x}^0 \) the approximation at iteration \( i \) is

\[
\hat{x}^i = \hat{x}^0 + \Psi \Phi^T (y - \Phi \Psi^{-1} \hat{x}^i)
\]  

(2)

\[
\hat{x}^i = \begin{cases} 
\hat{x}^i, & \| \hat{x}^i \| \geq \lambda^i \\
0, & \text{otherwise.}
\end{cases}
\]  

(3)

where \( \lambda^i \) is a threshold at each iteration, and \( \Phi \Phi^T = I \) [14]. According to the introduction above, we can find that despite many CS algorithms appearing in the literature, there are still many challenges in compressive sampling to approximate a signal. On one hand, in most methods a column or row of an image is normally viewed as a vector, and so the local 2D spatial image information is ignored. All the block compressive sensing methods mentioned above can achieve good performance, but they can still be improved. On the other hand, some classical methods, such as SPL and BCS_SPL, have good performance, but there are some parameters that need to be set by experience. Third, the computational cost for many methods, such as SP, GBP, CoSaMP, is unsatisfactory, and the time requirement increases substantially as the number of samples increases.

III. BLOCK COMPRESSION SENSING FOR SOLDER JOINT IMAGES WITH WAVELET PACKET THRESHOLDING

A. Block Compressive Sensing

In the classical methods, a column or row of an image is normally viewed as a vector. But in many applications the nonzero elements of sparse vectors tend to cluster in blocks [12]. Given an \( N_1 \times N_2 \) image, it is split into small blocks of size \( n_1 \times n_2 \), and it is transformed into a \( 1 \times n_1 n_2 \) vector. Let \( f_i \) represent the vectorized signal of the \( i \)-th block through raster scanning, \( i = 1, 2, \ldots, K \), and \( K = N_1 N_2/n_1 n_2 \). One is able to get an \( m \)-dimensional sampled vector \( y_b \) through the following linear transformation,

\[
y_b = \Phi_B f_i,
\]  

(4)

where \( \Phi_B \) is an \( n_1 n_2 \times n_1 n_2 \) measurement matrix which is constructed by Eq. 5.

\[
\Phi_{n_1 n_2} = \text{orth}(\text{randn}(n_1 n_2))
\]  

(5)

where \( \text{orth}(\cdot) \) is a function that generates an orthonormal basis for the input matrix, and \( \text{randn}(t) \) is a function for creating a random matrix of size \( t \times t \) whose entries are chosen independently from a normal distribution with zero mean and variance equal to \( \frac{1}{2} \) [28].

The block CS method is more efficient than the standard CS method as an \( m \times n_1 n_2 \) random matrix \( \Phi_B \) is generated for each image block. The small measurement matrix requires less memory storage and allows faster processing, while large data produces more accurate reconstruction.

One can learn from Eq. 4 that block compressive sensing is different from the common Compressive Sensing method which is based on using a column or row of the image to do the reconstruction. During block compressive sensing, an image is split into small blocks. This is because in most images the pixel values in a local patch are the same or similar. Especially in chip component solder joint images and gull-wing leaded solder joint images, the pixels in the area of the solder joint have similar values and the pixels in the background area have the same values. So during block compressive sensing, those pixels have a high probability to be split into the same block, and the orthogonal transformed image will have more sparsity than when using normal compressive sensing methods. This aids improving the the reconstruction result.

B. Wavelet Packet Thresholding

The Wavelet Packet Transform (WPT) is an efficient tool for signal analysis. The idea is exactly the same as those developed in the wavelet framework. Wavelet packet is a further generalization of wavelet analysis. The main difference is that the Wavelet Packet Transform offers a finer analysis, enabling finer control of partitioning the wavelet coefficients. The function groups are defined as follows:

\[
\begin{aligned}
y_{2n}(t) &= \sqrt{2} \sum_{k \in \mathbb{Z}} h_k(t) y_k(2t - k), \\
y_{2n+1}(t) &= \sqrt{2} \sum_{k \in \mathbb{Z}} g_k(t) y_k(2t - k),
\end{aligned}
\]  

(6)

where \( h(k) \) and \( g(k) \) are the wavelet filter coefficients in multi-resolution analysis. Specifically, when \( n = 0 \), Eq. 6 equals

\[
\begin{aligned}
y_0(t) &= \sqrt{2} \sum_{k \in \mathbb{Z}} h_k(t) y_0(2t - k), \\
y_1(t) &= \sqrt{2} \sum_{k \in \mathbb{Z}} g_k(t) y_0(2t - k),
\end{aligned}
\]  

(7)

where \( y_0(t) \) and \( y_1(t) \) correspond to the wavelet function and scaling function respectively.

After splitting, a vector of approximation coefficients and a vector of detail coefficients are obtained. So, the Wavelet Packet Transform can be more precise and provide comprehensive treatment of high-frequency signals and low-frequency signals which are very important in signal thresholding. We can use a complete binary tree to show its output in the following figure 1.

\( cD_j^h, cD_j^v, cD_j^d \) are details of the signal \( S \) in three orientations: horizontal, vertical, and diagonal, respectively. Wavelet packet
where $\lambda$ is the threshold value, $J$ is the total number of coefficients in each high frequency. The threshold to remove the reconstruction result in each iteration. Should have some connection with them. First, we define a higher frequency domains consist of noise, so the threshold generated by wavelet packet decomposition, and usually the high frequency coefficients and low frequency coefficients are generated by wavelet packet decomposition, and usually the higher frequency domains consist of noise, so the threshold should have some connection with them. First, we define a threshold to remove the reconstruction result in each iteration. So the threshold value efficiently. Assuming $J$ indicates the $J$-th wavelet packet decomposition, $K$ is the total number of coefficients of low frequency, and $i$ is the iteration number, the new control factor is defined as

$$\Gamma^i = \sqrt{\text{median} \left( \sum_{k=1}^{K} cA_i^j(k) \right)}$$ (11)

where $cA_i^j(k)$ are the $k$-th low frequency coefficients in the $J$-level wavelet packet decomposition in the $i$-th iteration. So the Control Factor is updated with the low frequency coefficients in each iteration.

### D. Algorithm

According to the introduction above, we now propose the BCS WP SPL algorithm whose details are shown in Algorithm 1.

In Algorithm 1, wpdec(·) is a function of wavelet package decomposition, and a db3 wavelet is used in our experiments. We split the image into blocks and each block is transformed into a one-dimensional data vector. We also used the Wiener filter to smooth the signal, and can choose different neighborhoods at different levels of the wavelet packet decomposition.

### E. Algorithm convergence analysis

In Algorithm 1, the discrete wavelet transform can be computed in $O(n)$ operations, and there are two transforms. So each iteration requires $O(2nk)$ iterations. Multiplication by the measurement matrix $\Phi$ is an intensive operation which requires $O(nk)$ operations. The hard-thresholding step is carried out independently in each iteration. It also requires $O(n)$ operations.

### IV. Results and Discussion

#### A. Sparsity Comparison

Some original solder joint images that will be used as test images are shown in figure 3. Given that $\hat{x}$ is defined as the...
Algorithm 1: Block Compressive Sensing based on wavelet package transform threshold

Input: An image $x$, a sparse signal transform matrix $\Psi \in R^{N \times N}$, a measurement matrix $\Phi \in R^{M \times N}$, $\Phi \Phi^T = I$, $M$ is the sample rate; $y = \Phi x$, wavelet transform level $J$.

Output: A reconstructed image $x$.

Procedure:

for each block $b$

$\hat{x}_b^0 = \Phi_b^T y_b$

end

$i = 0$; $r^0 = 1$; $r^{-1} = 0$.

while $|r^i - r^{i-1}| < 10^{-4}$ do

$\hat{x} = \text{Wiener}(x')$

for each block $b$

$\hat{x}_b^i = \hat{x}_b^i + \Phi^T_b (y_b - \Phi^T_b \hat{x}_b^i)$

$\check{x}_b^i = \Psi^T x'$

wpdec$(\check{x}, J)$

for each level $J$

for each subband $A_j \in \{cA_j\}$

end

for each block $b$

$\Gamma^*$ according to Eq. 11.

for each subband $D_j \in \{cD_j, cD'_j, cD''_j\}$

end

for each block $b$

$T$hreshold$(\hat{x}_b^i)$ according to Eq. 10

$\check{x}_b^i = \Psi^T \check{x}_b^i$

$\check{x}_b^{i+1} = \check{x}_b^{i} + \Phi^T_b (y_b - \Phi_b \check{x}_b^i)$

end

$r^{i+1} = \|x^{i+1} - \hat{x}_b^{i}\|_2$

$i = i + 1$

end

$x = x^{i+1}$

image after applying the orthogonal transform, the summed sparsity of its blocks is defined as

$$S_p = l_0^P (\check{x}_{i,j} \leq \epsilon),$$

(12)

where $\check{x}_{i,j}$ is the element at location $(i,j)$ in the $\check{x}$, and $l_0^P (\cdot)$ is a function defined in [15]. A comparison of image sparsity after applying the orthogonal transform is shown in table I.

One can see from table I that Block Compressive Sensing can achieve better sparsity than normal Compressive Sensing.

B. Experimental Comparison

In order to evaluate the quality of the reconstructed results, many researchers used the Peak Signal to Noise Rate (PSNR) and structural similarity (SSIM) to estimate the result in image processing [8]. In our study, PSNR and SSIM are used to compare the experimental results. The experiments were implemented on an Intel Core i5 with 2.70 GHz CPU. Since some methods require the image size to be a power of 2, we have cropped all the images to $256 \times 256$.

Now let us compare the proposed BCS_WP_SPL method with the popular methods CoSaMP [7], BCoSaMP [35] OMP [20], BOMP [12], FGB [36], BFGB, SP [6], GBP [25] and BCS SPL [24].

During BOMP, BCoSaMP, BFGB, the block size is set to $16 \times 16$. The reconstruction results based on popular methods with sample rate $u = 0.5$ ($M = N \times u = 128$) are shown in figure 4(a-i) and the reconstruction result based on BCS_WP_SPL with the same sample rate and the neighborhood in the Wiener filters $w = 3$, $Itr = 30$ iterations is shown in figure 4(j).

One can see that our method can achieve a better result than SP, OMP, GBP, CoSaMP, BOMP, BCoSaMP, FGB, BFGB, and BCS_SPL in figure 4. There are some block artifacts in figure 4(g,h). More PSNR and SSIM comparisons for a range of sampling rates are shown in table II.

From the figure and table above, one can see that the proposed BCS_WP_SPL approach obtains better results in terms of PSNR and SSIM than SP, GBP, CoSaMP, BOMP, OMP, BCoSaMP and BCS_SPL. The GBP method fails in image reconstruction when the sampling rate $u = 0.1$.

The runtime comparisons of different methods are shown in table III. SP, GBP, CoSaMP, BOMP, OMP and BCoSaMP cost more time with an increasing number of samples. OMP can achieve the fastest reconstruction. The BCS_SPL and BCS_WP_SPL methods require less time as the number of samples increases. BCS_WP_SPL costs more time than BCS_SPL, because BCS_WP_SPL costs extra time in wavelet packet decomposition.

C. Parameters Analysis

During BCS_WP_SPL, the Wiener filter is used to smooth the reconstruction result. We carried out more experiments with the image shown in figure 4(a) with different neighborhood sizes for the Wiener filters $(3 \times 3, 5 \times 5, 7 \times 7)$ and different wavelet packet decomposition levels $J = 2, 3$. The results are shown in table IV.

For both levels $J = 2, 3$, a $3 \times 3$ Wiener filter achieves better results in terms of PSNR and SSIM than the $5 \times 5$ and $7 \times 7$ Wiener filters. When the sampling rate $u < 0.5$ the proposed method based on 2 level wavelet packet decomposition achieves better results than 3 level wavelet packet in PSNR and SSIM. But when the sampling rate $u \geq 0.5$ the proposed method based on 3 level wavelet packet decomposition achieves better results than 2 level wavelet packet in PSNR and SSIM.

D. Small defect solder joint image reconstruction

For some challenging solder joint images with small defects, the proposed method can also achieve a better performance than other methods. A chip component defect solder joint
TABLE I: Sparsity comparison after applying the orthogonal transform for figure 3

<table>
<thead>
<tr>
<th>type</th>
<th>figure 3 (a)</th>
<th>figure 3 (b)</th>
<th>figure 3 (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Compressive Sensing</td>
<td>99.08%</td>
<td>99.39%</td>
<td>99.69%</td>
</tr>
<tr>
<td>Block Compressive Sensing</td>
<td>99.88%</td>
<td>99.39%</td>
<td>99.69%</td>
</tr>
</tbody>
</table>

(a) SP  (b) OMP  (c) GBP  (d) CoSaMP  (e) BOMP  (f) BCoSaMP  (g) FGb  (h) BFGB  (i) BCS_SPL  (j) BCS_WP_SPL

![Fig. 4: Reconstruction results based on different methods](image)

TABLE II: Quantitative comparison in PSNR and SSIM based on different methods for a range of sampling rates applied to figure 3 (a)

<table>
<thead>
<tr>
<th>Methods</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>8.83/0.051</td>
<td>12.36/0.110</td>
<td>16.55/0.218</td>
<td>18.83/0.326</td>
<td>20.33/0.399</td>
<td>21.94/0.489</td>
<td>23.02/0.545</td>
<td>24.20/0.613</td>
<td>25.14/0.655</td>
</tr>
<tr>
<td>OMP</td>
<td>9.51/0.120</td>
<td>13.83/0.185</td>
<td>17.47/0.315</td>
<td>19.70/0.448</td>
<td>21.97/0.544</td>
<td>23.82/0.667</td>
<td>25.31/0.735</td>
<td>26.93/0.808</td>
<td>27.89/0.737</td>
</tr>
<tr>
<td>GBP</td>
<td>0 /0</td>
<td>13.62/0.063</td>
<td>15.87/0.183</td>
<td>17.81/0.251</td>
<td>19.70/0.344</td>
<td>21.53/0.448</td>
<td>23.01/0.522</td>
<td>24.31/0.593</td>
<td>25.51/0.656</td>
</tr>
<tr>
<td>CoSaMP</td>
<td>8.61/0.031</td>
<td>10.62/0.063</td>
<td>15.87/0.183</td>
<td>17.81/0.251</td>
<td>19.70/0.344</td>
<td>21.53/0.448</td>
<td>23.01/0.522</td>
<td>24.31/0.593</td>
<td>25.51/0.656</td>
</tr>
<tr>
<td>BOMP</td>
<td>16.23/0.268</td>
<td>20.33/0.403</td>
<td>22.23/0.509</td>
<td>23.84/0.589</td>
<td>25.34/0.651</td>
<td>26.56/0.705</td>
<td>27.72/0.752</td>
<td>28.68/0.787</td>
<td>29.67/0.817</td>
</tr>
<tr>
<td>BCoSaMP</td>
<td>6.42/0.029</td>
<td>10.68/0.071</td>
<td>12.69/0.113</td>
<td>14.72/0.173</td>
<td>17.51/0.283</td>
<td>21.01/0.422</td>
<td>22.87/0.515</td>
<td>24.41/0.590</td>
<td>26.24/0.676</td>
</tr>
<tr>
<td>FGB</td>
<td>6.94/0.037</td>
<td>12.05/0.141</td>
<td>15.16/0.260</td>
<td>20.77/0.521</td>
<td>23.59/0.663</td>
<td>25.07/0.727</td>
<td>26.11/0.773</td>
<td>26.84/0.806</td>
<td>26.85/0.824</td>
</tr>
<tr>
<td>BFGB</td>
<td>13.63/0.239</td>
<td>22.32/0.625</td>
<td>25.02/0.690</td>
<td>26.29/0.736</td>
<td>27.19/0.766</td>
<td>27.90/0.787</td>
<td>28.69/0.810</td>
<td>29.34/0.827</td>
<td>29.98/0.843</td>
</tr>
<tr>
<td>BCS_SPL</td>
<td>26.96/0.732</td>
<td>29.36/0.808</td>
<td>29.74/0.814</td>
<td>32.45/0.885</td>
<td>33.74/0.909</td>
<td>35.29/0.933</td>
<td>36.87/0.951</td>
<td>39.01/0.968</td>
<td>42.12/0.984</td>
</tr>
<tr>
<td>BCS_WP_SPL</td>
<td>27.01/0.736</td>
<td>29.49/0.817</td>
<td>29.85/0.817</td>
<td>33.04/0.901</td>
<td>34.50/0.924</td>
<td>36.03/0.944</td>
<td>37.83/0.961</td>
<td>40.05/0.976</td>
<td>43.30/0.988</td>
</tr>
</tbody>
</table>

From the figure and table above, one can see that the proposed BCS_WP_SPL approach obtains better results in terms of PSNR and SSIM than SP, GBP, CoSaMP, BOMP, BCoSaMP, FGB, BFGB and BCS_SPL. The GBP method fails in image reconstruction when the sampling rate $u = 0.1$.

E. Different types of solder joint image experiment

We also experiment with different types of solder joint image. A chip component solder joint image and its reconstruction results are shown in figure 3(c).

During the BCS_WP_SPL, we set $J = 2$ and the Wiener filter neighborhood size 3. The reconstruction results when the sampling rate is $u = 0.5$ are shown as in figure 6.

We carry out more experiments with the image in figure 3(c) with different sampling rates $u = [0.1, 0.9]$. The results are shown in tables VI.

From table VI, one can see that the proposed approach obtains better results in terms of PSNR and SSIM than SP, GBP, CoSaMP, BOMP, OMP BCoSaMP and BCS_SPL. The GBP method fails in image reconstruction when the sampling rate $u = 0.1$.

Compared to BCS_SPL, the proposed approach achieves better results in terms of PSNR and SSIM than BCS_SPL at most sampling rates. When the sampling rate $u = 0.2, 0.4$, BCS_SPL can achieve a better result than BCS_WP_SPL, but the proposed approach can achieve a better result in terms of SSIM than BCS_SPL. This means BCS_WP_SPL has a better performance in retaining image structure than BCS_SPL.
TABLE III: Runtime comparison based on different methods for a range of sampling rates applied to figure 3(a)

<table>
<thead>
<tr>
<th>Methods</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>0.6</td>
<td>1.5</td>
<td>10.5</td>
<td>16.2</td>
<td>22.3</td>
<td>40.3</td>
<td>57.3</td>
<td>66.4</td>
<td>100.3</td>
</tr>
<tr>
<td>OMP</td>
<td>0.2</td>
<td>0.5</td>
<td>2.8</td>
<td>6.4</td>
<td>6.6</td>
<td>10.0</td>
<td>13.3</td>
<td>14.6</td>
<td>21.3</td>
</tr>
<tr>
<td>GBP</td>
<td>0</td>
<td>11.9</td>
<td>17.4</td>
<td>25.4</td>
<td>39.7</td>
<td>50.8</td>
<td>63.8</td>
<td>77.1</td>
<td>95.9</td>
</tr>
<tr>
<td>CoSaMP</td>
<td>0.6</td>
<td>5.0</td>
<td>12.4</td>
<td>26.8</td>
<td>37.0</td>
<td>59.4</td>
<td>78.7</td>
<td>124.7</td>
<td>157.2</td>
</tr>
<tr>
<td>BOMP</td>
<td>5.0</td>
<td>4.5</td>
<td>10.3</td>
<td>12.3</td>
<td>13.4</td>
<td>16.6</td>
<td>19.3</td>
<td>25.2</td>
<td>23.3</td>
</tr>
<tr>
<td>BCoSaMP</td>
<td>0.6</td>
<td>4.9</td>
<td>13.8</td>
<td>21.6</td>
<td>38.6</td>
<td>62.2</td>
<td>81.7</td>
<td>122.2</td>
<td>136.2</td>
</tr>
<tr>
<td>FGB</td>
<td>4.1</td>
<td>21.5</td>
<td>22.4</td>
<td>29.2</td>
<td>30.9</td>
<td>35.3</td>
<td>41.9</td>
<td>40.1</td>
<td>40.9</td>
</tr>
<tr>
<td>BFGB</td>
<td>4.5</td>
<td>22.5</td>
<td>23.5</td>
<td>30.2</td>
<td>31.8</td>
<td>36.4</td>
<td>42.8</td>
<td>41.1</td>
<td>41.1</td>
</tr>
<tr>
<td>BCS_SPL</td>
<td>10.3</td>
<td>8.7</td>
<td>7.3</td>
<td>8.4</td>
<td>2.6</td>
<td>2.1</td>
<td>2.1</td>
<td>1.7</td>
<td>1.4</td>
</tr>
<tr>
<td>BCS_WP_SPL</td>
<td>29.9</td>
<td>22.6</td>
<td>20.5</td>
<td>21.6</td>
<td>19.4</td>
<td>5.6</td>
<td>4.9</td>
<td>4.4</td>
<td>4.1</td>
</tr>
</tbody>
</table>

TABLE IV: Quantitative comparison in PSNR and SSIM based on different Wiener filters neighborhoods for figure 3(a)

<table>
<thead>
<tr>
<th>Methods</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP</td>
<td>0.1</td>
<td>0.5</td>
<td>1.2</td>
<td>2.5</td>
<td>5.0</td>
<td>8.0</td>
<td>12.0</td>
<td>17.0</td>
<td>22.0</td>
</tr>
<tr>
<td>BOMP</td>
<td>0.2</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
<td>4.0</td>
<td>6.0</td>
<td>9.0</td>
<td>13.0</td>
<td>18.0</td>
</tr>
<tr>
<td>GBP</td>
<td>0.3</td>
<td>0.7</td>
<td>1.4</td>
<td>2.8</td>
<td>5.6</td>
<td>8.4</td>
<td>12.6</td>
<td>18.2</td>
<td>24.0</td>
</tr>
<tr>
<td>BCoSaMP</td>
<td>0.4</td>
<td>0.8</td>
<td>1.6</td>
<td>3.2</td>
<td>6.4</td>
<td>9.6</td>
<td>14.4</td>
<td>21.0</td>
<td>28.0</td>
</tr>
<tr>
<td>FGB</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
<td>4.0</td>
<td>8.0</td>
<td>12.0</td>
<td>18.0</td>
<td>24.0</td>
<td>30.0</td>
</tr>
<tr>
<td>BFGB</td>
<td>0.6</td>
<td>1.2</td>
<td>2.4</td>
<td>4.8</td>
<td>9.6</td>
<td>14.4</td>
<td>21.0</td>
<td>28.0</td>
<td>36.0</td>
</tr>
<tr>
<td>BCS_SPL</td>
<td>0.7</td>
<td>1.4</td>
<td>2.8</td>
<td>5.6</td>
<td>11.2</td>
<td>17.6</td>
<td>26.4</td>
<td>36.0</td>
<td>46.8</td>
</tr>
<tr>
<td>BCS_WP_SPL</td>
<td>0.8</td>
<td>1.6</td>
<td>3.2</td>
<td>6.4</td>
<td>12.8</td>
<td>19.2</td>
<td>28.8</td>
<td>38.4</td>
<td>49.6</td>
</tr>
</tbody>
</table>

F. Soft-thresholding experiment

The proposed algorithm uses hard thresholding to filter a transformed signal. [11] has proved that soft thresholding cannot be used to solve the problem very well because the terms $\Phi^*F^{-1}$ in Eq. 1 are not separable in the $l_1$ optimization. However, we also perform experiments to evaluate the use of soft-thresholding to filter the transformed signal. $S(x, \lambda)$ is defined as a soft-thresholding operator in Eq. 13.

$$S(x, \lambda) = \text{sign}(x)(|x| - \lambda)_+ = \begin{cases} x - r, & \text{if } x > 0 \text{ and } \lambda < |x|, \\ x + r, & \text{if } x < 0 \text{ and } \lambda < |x|, \\ 0, & \text{if } \lambda \geq |x|. \end{cases}$$

where $x$ is the transformed signal and $\lambda$ is the thresholding value. Figure 4(a), figure 5 and figure 6(a) show more experiments with soft-thresholding. During BCS_WP_SPL, we set $J = 2$ and the Wiener filter neighborhood size 3. The reconstruction results with sampling rate $u = 0.5$ are shown in figure 7.

We carry out more experiments for images in figure 3 with different sampling rates $u = [0.1, 0.9]$. The results are shown in tables VII.

Comparing table VII with the results in the preceding tables shows that, for BCS_SPL and BCS_WP_SPL, hard-thresholding achieves better results than soft-thresholding.

G. Dataset experiment

We have created a dataset of solder joint images to enable more thorough experimentation. The test dataset has 180 images, and consists of 3 different types of solder joints: gull-wing leaded solder joint, Ball Grid Array (BGA) solder joint, chip component solder joint. For each type there are 30 acceptable images and 30 defective images which include some challenging samples with small defects. The details of the solder joint image dataset are shown in table VIII.

Some images are shown in figure 8. During the BCS_WP_SPL, we set $J = 2$ and Wiener filter neighborhood
size 3. The reconstruction results with different sample numbers based on different methods are shown in table IX.

From table IX, one can see that the proposed approach obtains better results in terms of PSNR and SSIM than SP, GBP, CoSaMP, BOMP, OMP and BCoSaMP. When the sample rate $u \leq 0.2$, BCS_WP_SPL achieves similar result with BCS_SPL, but when $u \leq 0.3$ the PSNR value is improved more than 0.5 at each sample rate, and the SSIM value is also improved.

V. CONCLUSION

This paper proposes a wavelet packet thresholding (BCS_WP_SPL) approach on the basis of wavelet packet coefficients of the image. Experiments reveal that

- Wavelet packet decomposition divides the frequency space into various parts and allows better frequency localization of signals. We define a threshold which only relates to the higher frequency coefficients to remove the noise in the reconstruction result at each iteration. We define a new control factor $\Gamma$ which is based on the wavelet packet low frequency coefficients. The new control factor is updated by the wavelet packet low frequency coefficients in each iteration, so it can efficiently remove the noise and avoid block artifacts.

- The proposed algorithm can achieve better results ac-
TABLE IX: Average PSNR and SSIM comparison based on different methods for a range of sampling rates applied to the solder joint image dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>8.746/0.079</td>
<td>12.599/0.159</td>
<td>15.88/0.249</td>
<td>18.46/0.333</td>
<td>20.55/0.404</td>
<td>22.57/0.477</td>
<td>24.30/0.537</td>
<td>25.64/0.585</td>
<td>26.89/0.631</td>
</tr>
<tr>
<td>OMP</td>
<td>13.42/0.166</td>
<td>17.25/0.267</td>
<td>19.76/0.361</td>
<td>21.81/0.439</td>
<td>23.57/0.504</td>
<td>25.14/0.563</td>
<td>26.49/0.612</td>
<td>27.61/0.651</td>
<td>28.58/0.684</td>
</tr>
<tr>
<td>GBP</td>
<td>0/0</td>
<td>14.14/0.239</td>
<td>16.90/0.335</td>
<td>18.87/0.406</td>
<td>20.44/0.460</td>
<td>21.86/0.509</td>
<td>22.31/0.552</td>
<td>23.21/0.613</td>
<td>24.24/0.651</td>
</tr>
<tr>
<td>CoSaMP</td>
<td>7.49/0.055</td>
<td>10.89/0.113</td>
<td>14.55/0.199</td>
<td>17.47/0.290</td>
<td>19.95/0.369</td>
<td>22.38/0.457</td>
<td>24.37/0.529</td>
<td>25.94/0.590</td>
<td>27.32/0.645</td>
</tr>
<tr>
<td>BOMP</td>
<td>17.81/0.551</td>
<td>20.95/0.646</td>
<td>23.14/0.709</td>
<td>24.95/0.761</td>
<td>26.46/0.802</td>
<td>27.92/0.838</td>
<td>29.21/0.866</td>
<td>30.31/0.887</td>
<td>31.31/0.904</td>
</tr>
<tr>
<td>BCoSaMP</td>
<td>8.39/0.067</td>
<td>10.99/0.115</td>
<td>13.62/0.166</td>
<td>16.22/0.234</td>
<td>18.47/0.306</td>
<td>20.80/0.395</td>
<td>23.12/0.480</td>
<td>24.83/0.547</td>
<td>26.25/0.604</td>
</tr>
<tr>
<td>FGB</td>
<td>9.56/0.137</td>
<td>17.96/0.476</td>
<td>20.60/0.646</td>
<td>22.42/0.734</td>
<td>23.47/0.782</td>
<td>25.15/0.821</td>
<td>26.46/0.852</td>
<td>27.31/0.876</td>
<td>28.55/0.898</td>
</tr>
<tr>
<td>BFGB</td>
<td>13.41/0.166</td>
<td>21.23/0.615</td>
<td>23.02/0.768</td>
<td>24.35/0.801</td>
<td>25.26/0.827</td>
<td>27.40/0.876</td>
<td>28.21/0.891</td>
<td>29.14/0.913</td>
<td>29.87/0.929</td>
</tr>
<tr>
<td>BCS_SPL</td>
<td>25.58/0.821</td>
<td>29.51/0.888</td>
<td>31.70/0.915</td>
<td>33.73/0.939</td>
<td>35.68/0.956</td>
<td>37.52/0.969</td>
<td>39.58/0.979</td>
<td>42.19/0.988</td>
<td>45.87/0.994</td>
</tr>
<tr>
<td>BCS_WP_SPL</td>
<td>25.63/0.822</td>
<td>29.60/0.890</td>
<td>32.33/0.918</td>
<td>34.40/0.941</td>
<td>36.39/0.959</td>
<td>38.29/0.971</td>
<td>40.26/0.990</td>
<td>43.06/0.990</td>
<td>46.73/0.995</td>
</tr>
</tbody>
</table>

Fig. 7: Solder joint image reconstruction results with soft-thresholding

With different levels $J = 2, 3$, a $3 \times 3$ Wiener filter achieves better results according to PSNR and SSIM than classical algorithms for reconstruction of images of different types of solder joints.

- With different levels $J = 2, 3$, a $3 \times 3$ Wiener filter achieves better results according to PSNR and SSIM than classical algorithms.
- With different levels $J = 2, 3$, a $3 \times 3$ Wiener filter achieves better results according to PSNR and SSIM than classical algorithms.
- With different levels $J = 2, 3$, a $3 \times 3$ Wiener filter achieves better results according to PSNR and SSIM than classical algorithms.
- With different levels $J = 2, 3$, a $3 \times 3$ Wiener filter achieves better results according to PSNR and SSIM than classical algorithms.

By doing tests in the solder joint image dataset which contains acceptable images and defective images of different solder joint types, the proposed algorithm can achieve better results according to PSNR and SSIM than classical algorithms. With an increasing sample rate, the proposed method improves the reconstruction result.

Fig. 8: Some examples in the solder joint image dataset

In a future study, more relationships between wavelet packet coefficients of images and image compressive sensing reconstruction will be researched, and we will test more types of solder joint images. We will also test more natural images with the proposed method.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (61503128), Science and Technol-
ogy Plan Project of Hunan Province (2016TP102), Scientific Research Fund of Hunan Provincial Education Department (16C0226,18A333), Hengyang guided science and technology projects and Application-oriented Special Disciplines (Hengeka [2018]60-31), Double First-Class University Project of Hunan Province (Xiangjiaotong [2018]469), Hunan Province Special Funds of Central Government for Guiding Local Science and Technology Development (2018CT5001) and Subject Group Construction Project of Hengyang Normal University (18XKQ02). We would like to thank NVIDIA for the GPU donation.

REFERENCES


Hui-Huang Zhao received his Ph.D. degree in 2010 from XiDian University. He was a Sponsored Researcher in the School of Computer Science and Informatics, Cardiff University. Now he is an Associate Professor in the College of Computer Science and Technology, Hengyang Normal University. His main research interests include Solder Joint Inspection, Compressive Sensing, Machine Learning, and Image Processing.
Paul L. Rosin received the B.Sc. degree in Computer Science and Microprocessor Systems in 1984 from Strathclyde University, Glasgow, and the Ph.D. degree in Information Engineering from City University, London in 1988. He is a full professor in the School of Computer Science and Informatics, Cardiff University. His main research interests include Non-Photorealistic Rendering, Mesh Processing, and Computer Vision.

Yu-Kun Lai received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Reader of Visual Computing in the School of Computer Science and Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial board of The Visual Computer.