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.

Hui-huang Zhao is with College of Computer Science and Technology, Hengyang Normal University, e-mail:happyday.huihuang@gmail.com. Paul L. Rosin and Yu-Kun Lai are with School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff, CF24 3AA, United Kingdom.

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 (DR^2-Net). DR^2-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 l_1 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_WP_SPL 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:

$$\begin{cases} \text{minimize} \parallel \hat{x} \parallel_1 \\ \text{subject to} : y = \Phi\Psi^{-1}\hat{x} \end{cases} \tag{1}$$

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

$$\check{x}^i = \hat{x}^i + \Psi\Phi^T(y - \Phi\Psi^{-1}\hat{x}^i) \tag{2}$$

$$\check{x}^i = \begin{cases} \check{x}^i, & \left|\check{x}^i\right| \geq \lambda^i \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where λ^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 Compressive 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 $1 \times n_1n_2$ vector. Let f_i represent the vectorized signal of the i-th block through raster scanning, i=1, 2, …, K, and $K = N_1N_2/n_1n_2$. One is able to get an m-dimensional sampled vector y_B through the following linear transformation,

$$y_B = \Phi_B f_i, \tag{4}$$

where Φ_B is an $n_1n_2 \times n_1n_2$ measurement matrix which is constructed by Eq. 5.

$$\Phi_{n_1n_2} = orth((randn(n_1n_2)) \tag{5}$$

where $orth(\cdot)$ is a function that generates an orthonormal basis for the input matrix, and $randn(t)$ 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 $\frac{1}{t}$ [28].

The block CS method is more efficient than the standard CS method as an $m \times n_1n_2$ random matrix Φ_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 Transfer Threshold

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:

$$\begin{cases} y_{2n}(t) = \sqrt{2}\sum_{k\in Z} h_k(t)y_k(2t-k), \\ y_{2n+1}(t) = \sqrt{2}\sum_{k\in Z} g_k(t)y_k(2t-k). \end{cases} \tag{6}$$

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

$$\begin{cases} y_0(t) = \sqrt{2}\sum_{k\in Z} h_k(t)y_0(2t-k), \\ y_1(t) = \sqrt{2}\sum_{k\in Z} g_k(t)y_0(2t-k). \end{cases} \tag{7}$$

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

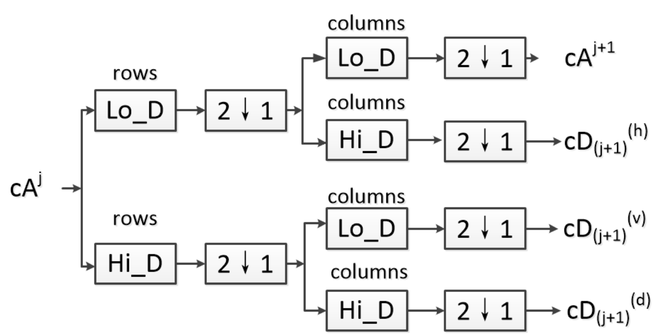


Fig. 1: 2-Dimensional Discrete Wavelet Packet Transform

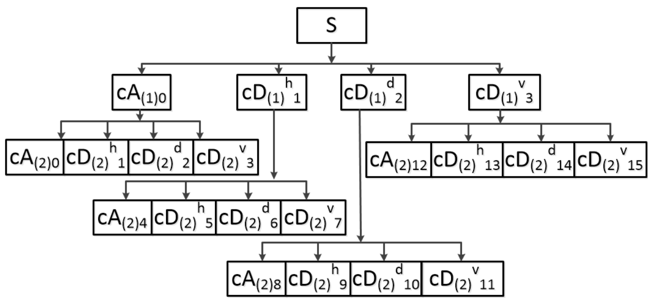


Fig. 2: A two-dimensional wavelet packet tree containing two layers

decomposes the signal at high frequency (cD_j^h, cD_j^v, cD_j^d) and low frequency (cA_j).

In general, wavelet packet decomposition divides the frequency space into various parts and allows better frequency localization of signals. So, for 2 level decomposition, the two-dimensional wavelet packet tree has the form in figure 2.

The root of the tree is the original image. The next level of the tree is the result of one step of the wavelet packet decomposition.

We can see from the introduction above, that wavelet packet transfer has special abilities to achieve higher discrimination by analyzing the higher frequency domains of a signal During our approach, we deal with the higher frequency and lower frequency of the reconstruction signal separately. 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. J indicates the J-th wavelet packet decomposition, and K is the total number of coefficients in each high frequency. The threshold value λ is defined as

$$\lambda = \Gamma \sigma_J \tag{8}$$

where Γ is defined as a control factor which connects with low frequency coefficients to manage convergence and remove the noise, and σ_J is a median estimator of the standard deviation, and is defined as

$$\sigma_J = \frac{median(\left|\sum_{k=1}^{K}(cD_J^h(k) + cD_J^v(k) + cD_J^d(k)\right|)}{0.6745} \tag{9}$$

where cD_J^h, cD_J^v, and cD_J^d are the k-th high frequency coefficients in the J-level wavelet packet decomposition in the horizontal, vertical, and diagonal orientations respectively. 0.6745 is a value making the median an unbiased estimator for the normal distribution [19]. For each iteration result x, we deal with it as

$$Threshold(x) = \begin{cases} x(k), & |x(k)| \geq \lambda \\ 0, & |x(k)| < \lambda. \end{cases} \tag{10}$$

where $x(k)$ is the k-th element in x.

C. The Control Factor with Wavelet Packet Coefficients

As we mentioned earlier, there is a control factor in Eq. 8. In general, the control factor is based on experience and it is constant. In this paper, we define the control factor Γ which connects with the wavelet packet coefficients. The new control factor is updated using the wavelet packet low frequency coefficients in each iteration, so it can control 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

$$\Gamma^i = \sqrt{median\left(\sum_{k=1}^{K} cA_J^i(k)\right)} \tag{11}$$

where $cA_J^i(k)$ 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, $wpdec(\cdot)$ 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 Φ 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
  $x_b^0 = \Phi_b^T y_b$
end
$i = 0; r^0 = 1; r^{-1} = 0.$
while $\left|r^i - r^{i-1}\right| < 10^{-4}$ **do**
  $\hat{x}^i$=Wiener(x^i)
  for each block b
    $\tilde{x}_b^i = \hat{x}_b^i + \Phi_b^T(y_b - \Phi_b^T \hat{x}_b^i)$
  $\check{x}^i = \Psi\tilde{x}^i$
  $wpdec(\check{x}, J)$
  for each level J
    for each subband $A_J \in \{cA_J\}$
      for each block b
        Γ^i according to Eq. 11.
    for each subband $D \in \{cD_J^h, cD_J^v, cD_J^d\}$
      for each block b
        $Threshold(\check{x}_b^i)$ according to Eq. 10
  $\hat{\hat{x}}^i = \Psi^{-1}\check{x}^i$
  for each block b
    $x_b^{i+1} = \hat{\hat{x}}_b^i + \Phi_b^T(y_b - \Phi_b \hat{\hat{x}}_b^i)$
  end
  $r^{i+1} = \| x^{i+1} - \hat{\hat{x}}^i \|_2$
  $i = i + 1$
end
$x = x^{i+1}$

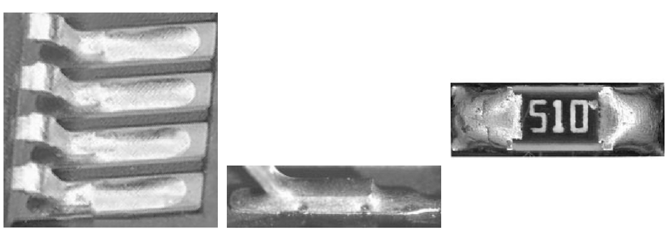


Fig. 3: Solder images

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

$$S_p = l_\varepsilon^0\left(\hat{x}_{i,j} \le \varepsilon\right), \tag{12}$$

where $\hat{x}_{i,j}$ is the element at location (i, j) in the $\hat{x}$, and $l_\varepsilon^0(\cdot)$ 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 256×256.

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 16×16. 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 (3×3, 5×5, 7×7) and different wavelet packet decomposition levels $J = 2, 3$. The results are shown in table IV.

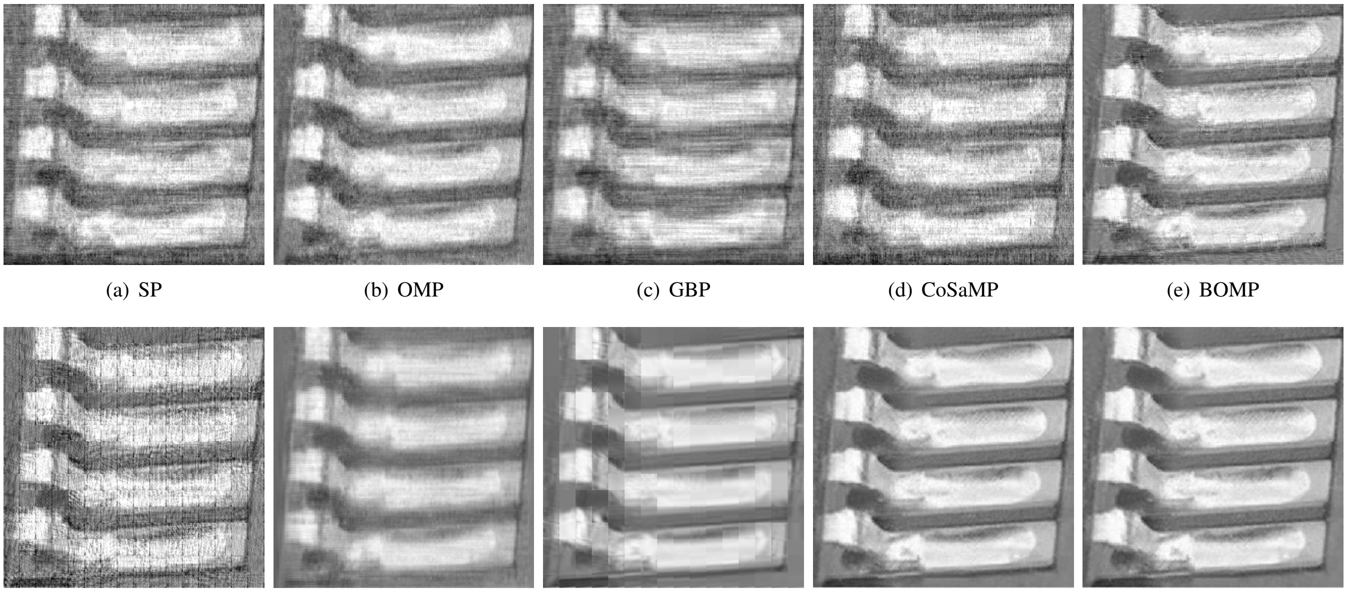
For both levels $J = 2, 3$, a 3×3 Wiener filter achieves better results in terms of PSNR and SSIM than the 5×5 and 7×7 Wiener filters. When the sampling rate $u < 0.5$ the proposed method based on 2 level wavelet packet decomposition achieves better results than 3 level wavelet packet in PSNR and SSIM.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

type	figure 3 (a)	figure 3 (b)	figure 3 (c)
Normal Compressive Sensing	99.08%	95.88%	98.32%
Block Compressive Sensing	99.39%	99.39%	99.69%



(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

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

Methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
SP	8.83/0.051	12.76/0.110	16.55/0.218	18.83/0.326	20.33/0.399	21.94/0.489	23.02/0.545	24.20/0.613	25.14/0.655
OMP	9.51/0.120	18.06/0.251	19.68/0.335	21.10/0.400	22.46/0.476	24.46/0.575	25.81/0.646	26.93/0.698	27.89/0.737
GBP	0 /0	13.83/0.185	17.47/0.315	19.41/0.419	20.59/0.475	21.97/0.544	22.86/0.588	23.62/0.625	24.47/0.657
CoSaMP	8.61/0.031	10.62/0.063	15.87/0.183	17.81/0.251	19.70/0.344	21.53/0.448	23.01/0.522	24.31/0.593	25.51/0.656
BOMP	16.23/0.268	20.33/0.403	22.23/0.509	23.84/0.589	25.34/0.651	26.56/0.705	27.72/0.752	28.68/0.787	29.67/0.817
BCoSaMP	6.42/0.029	10.68/0.071	12.69/0.113	14.72/0.173	17.51/0.283	21.01/0.422	22.87/0.515	24.41/0.590	26.24/0.676
FGB	6.94/0.037	12.05/0.141	15.16/0.260	20.77/0.521	23.59/0.663	25.07/0.727	26.11/0.773	26.84/0.806	26.85/0.824
BFGB	13.63/0.239	23.23/0.625	25.02/0.690	26.29/0.736	27.19/0.766	27.90/0.787	28.69/0.810	29.34/0.827	29.98/0.843
BCS_SPL	26.96/0.732	29.36/0.808	29.74/0.814	32.45/0.885	33.78/0.909	35.28/0.933	36.87/0.951	39.01/0.968	42.12/0.984
BCS_WP_SPL	**27.01/0.736**	**29.49/0.8175**	**29.85/0.817**	**33.04/0.901**	**34.50/0.924**	**36.03/0.944**	**37.83/0.961**	**40.05/0.976**	**43.30/0.988**

image which has two small concavities is shown in figure 3(b). Its reconstruction results are shown in figure 5.

During the BCS_WP_SPL, we set $J = 2$ and the Wiener filter neighborhood size 3.

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 5. There are some block artifacts in figure 5(g,h). More PSNR and SSIM comparisons for a range of sampling rates are shown in table V.

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 $u = 0.1$.

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. BCS_SPL achieves the second best results. With increasing number of samples, the proposed approach gets better reconstruction results (unlike some other methods).

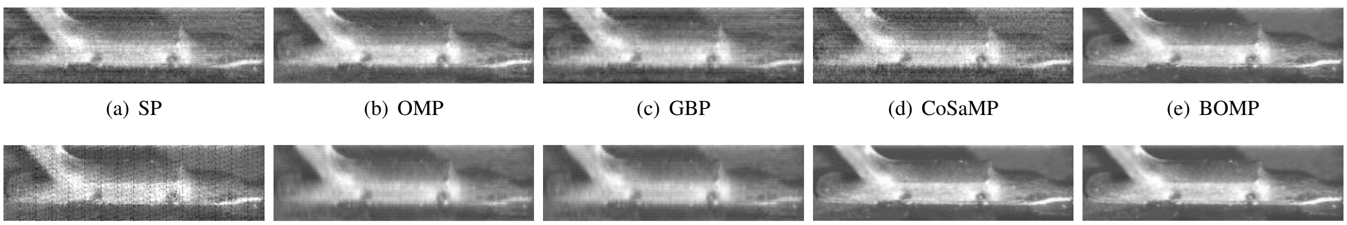
Compared to BCS_SPL, the proposed approach achieves better results in terms of PSNR and SSIM than BCS_SPL at most sampling rates. When the sampling rate $u = 0.2, 0.4$, BCS_SPL can achieve a better result than BCS_WP_SPL, but the proposed approach can achieve a better result in terms of SSIM than BCS_SPL. 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)

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

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

level	size	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
J=2	3×3	**27.01/0.736**	**29.49/0.8175**	**29.85/0.817**	**33.04/0.901**	34.50/0.924	36.03/0.944	37.83/0.961	40.05/0.976	43.30/0.988
J=2	5×5	24.75/0.665	26.31/0.692	29.15/0.812	30.83/0.857	32.78/0.893	34.51/0.923	36.11/0.948	38.53/0.968	42.12/0.985
J=2	7×7	24.25/0.625	26.43/0.702	27.34/0.752	28.95/0.805	31.15/0.853	33.35/0.906	34.25/0.927	36.91/0.957	40.59/0.980
J=3	3×3	26.35/ 0.705	26.64/ 0.707	28.19/ 0.763	32.53/ 0.887	**34.52/ 0.926**	**36.08/ 0.946**	**37.90/ 0.963**	**40.13/ 0.977**	**43.39/ 0.989**
J=3	5×5	25.16/ 0.674	27.59/ 0.756	29.31/ 0.815	31.05/ 0.860	32.58/ 0.894	34.30/ 0.924	36.26/ 0.948	38.68/ 0.969	42.23/ 0.985
J=3	7×7	23.49/ 0.608	26.36/ 0.698	28.05/ 0.762	30.07/ 0.829	31.53/ 0.867	32.63/ 0.887	34.50/ 0.930	37.21/ 0.959	40.89/ 0.981



(a) SP (b) OMP (c) GBP (d) CoSaMP (e) BOMP

(f) BCoSaMP (g) FGb (h) BFGB (i) BCS_SPL (j) BCS_WP_SPL

Fig. 5: A small defect solder joint image reconstruction results

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

Methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
SP	12.78/0.102	16.241/0.148	19.73/0.271	20.72/0.317	22.29/0.382	23.34/0.441	24.41/0.492	25.02/0.513	25.89/0.564
OMP	14.89/0.163	21.91/0.389	25.08/0.535	26.58/0.607	28.00/0.668	29.14/0.720	30.62/0.778	31.62/0.807	32.64/0.836
GBP	0/0	17.35/0.240	20.79/0.414	21.52/0.449	22.44/0.49223.52/	0.54324.33/	0.57324.88/	0.598	25.55/0.642
CoSaMP	12.864/0.074	14.93/0.096	18.22/0.183	19.43/0.230	20.89/0.287	22.65/0.375	23.95/0.434	24.89/0.486	26.36/0.575
BOMP	22.13/0.482	24.359/0.579	26.56/0.666	28.81/0.751	30.66/0.813	32.48/0.859	34.12/0.894	35.48/0.915	36.5/0.930
BCoSaMP	9.98/0.036	13.21/0.075	16.20/0.131	17.93/0.18	20.19/0.274	22.69/0.406	25.45/0.534	26.72/0.592	28.17/0.657
FGB	12.54/0.196	20.87/0.605	23.76/0.753	25.59/0.810	26.49/0.838	27.45/0.868	28.26/0.890	28.69/0.897	29.74/0.906
BFGB	16.73/0.501	24.74/0.701	26.73/0.744	28.27/0.781	29.27/0.805	30.04/0.823	30.92/0.843	31.62/0.857	32.36/0.872
BCS_SPL	31.35/0.859	34.79/0.914	36.95/0.942	39.04/0.960	41.02/0.973	43.04/0.982	45.46/0.989	48.48/0.994	52.15/0.997
BCS_WP_SPL	**31.56/0.879/**	**34.98/0.925/**	**40.24/0.952/**	**39.38/0.971/**	**41.34/0.978/**	**43.36/0.984/**	**45.76/0.991/**	**48.82/0.996/**	**52.38/0.998**

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 $\Phi\Psi^{-1}$ 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.

$$S(x,\lambda) = sign(x)(|x|-\lambda)_+ = \begin{cases} x-r, & \text{if } x>0 \text{ and } \lambda<|x|, \\ x+r, & \text{if } x<0 \text{ and } \lambda<|x|, \\ 0, & \text{if } \quad \lambda \geq |x|. \end{cases} \tag{13}$$

where x is the transformed signal and λ 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.

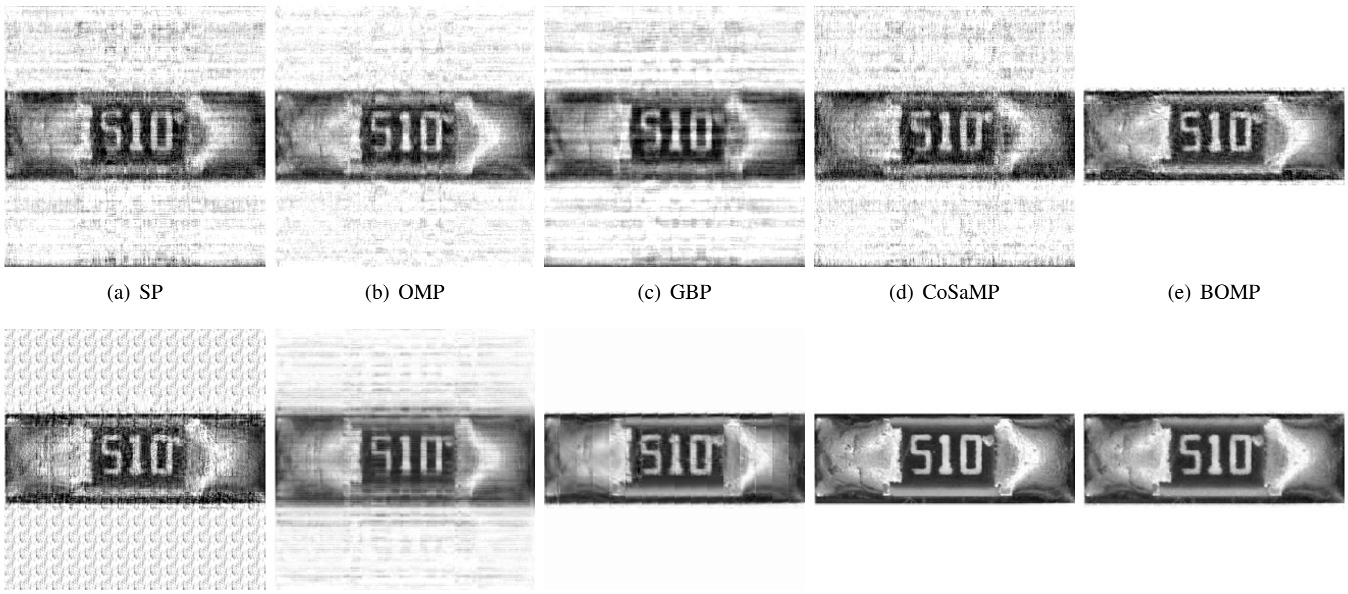
We carry out more experiments for images in figure 3 with different sampling rates $u=[0.1,0.9]$. The results are shown in tables VII.

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



(a) SP (b) OMP (c) GBP (d) CoSaMP (e) BOMP

(f) BCoSaMP (g) FGb (h) BFGb (i) BCS_SPL (j) BCS_WP_SPL

Fig. 6: Reconstruction results of a chip component solder joint image

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

Methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
SP	4.59/0.011	8.50/0.043	12.31/0.108	14.16/0.148	16.07/0.204	18.42/0.275	20.34/0.347	21.53/0.396	23.32/0.479
OMP	12.01/0.076	14.20/0.123	16.30/0.188	17.89/0.245	19.44/0.303	20.79/0.350	22.62/0.420	23.75/0.468	25.05/0.527
GBP	0 /0	9.59/0.067	13.35/0.180	15.34/0.230	16.72/0.272	18.10/0.327	19.05/0.359	20.32/0.409	21.18/0.441
CoSaMP	4.33/0.014	6.73/0.025	10.11/0.060	13.29/0.111	15.77/0.181	18.20/0.260	20.53/0.337	22.21/0.414	23.42/0.461
BOMP	18.00/0.697	19.73/0.744	22.17/0.789	23.34/0.815	25.44/0.852	26.75/0.876	27.72/0.895	28.96/0.912	30.19/0.929
BCoSaMP	2.24/0.025	6.96/0.066	12.64/0.122	14.94/0.162	16.95/0.211	18.23/0.249	21.67/0.366	23.25/0.435	24.56/0.489
FGB	3.99/0.040	7.90/0.079	9.94/0.101	14.63/0.254	18.94/0.511	22.50/0.732	24.70/0.816	25.14/0.835	25.71/0.845
BFGB	9.90/0.126	22.48/0.828	24.48/0.867	26.11/0.892	27.51/0.911	28.55/0.924	30.05/0.938	30.99/0.947	32.13/0.956
BCS_SPL	22.01/0.821	25.75/0.895	28.17/0.928	30.70/0.951	32.76/0.964	36.36/0.979	37.17/0.983	39.62/0.989	43.16/0.995
BCS_WP_SPL	**23.88/0.848**	**26.62/ 0.916**	**29.10/0.936**	**31.47/0.961**	**33.74/0.968**	**36.64/0.981**	**38.07/0.985**	**40.69/0.991**	**44.64/0.996**

TABLE VII: Quantitative comparison using PSNR and SSIM with soft-thresholding for figure 3

level	size	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
BCS_SPL	figure 3 (a)	16.91/0.427	26.82/0.905	28.92/0.933	31.37/0.953	33.40/0.966	35.60/0.976	37.71/0.984	40.10/0.990	43.60/0.995
BCS_SPL	figure 3 (b)	26.43/0.722	32.25/0.865	34.49/0.904	36.38/0.930	38.24/0.950	40.35/0.966	42.68/0.979	45.53/0.988	49.48/0.995
BCS_SPL	figure 3 (c)	21.62/0.500	25.66/0.720	28.41/0.781	30.04/0.827	31.36/0.860	32.98/0.895	34.62/0.922	36.88/0.950	40.01/0.974
BCS_WP_SPL	figure 3 (a)	22.91/0.827	27.01/0.899	29.61/0.930	31.92/0.950	33.84/0.963	35.88/0.974	38.13/0.983	40.65/0.990	44.11/0.995
BCS_WP_SPL	figure 3 (b)	29.16/0.808	32.34/0.866	34.53/0.905	36.53/0.933	38.40/0.952	40.45/0.968	42.79/0.980	45.66/0.989	49.62/0.996
BCS_WP_SPL	figure 3 (c)	22.71/0.650	25.83/0.727	28.53/0.785	30.11/0.829	31.44/0.862	32.99/0.896	34.65/0.923	36.87/0.953	40.07/0.975

TABLE VIII: Solder joint image dataset

type	acceptable	defective	total
Gull-wing leaded solder joint	30	30	60
Ball Grid Array Solder Joint	30	30	60
Chip component solder joint	30	30	60

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.

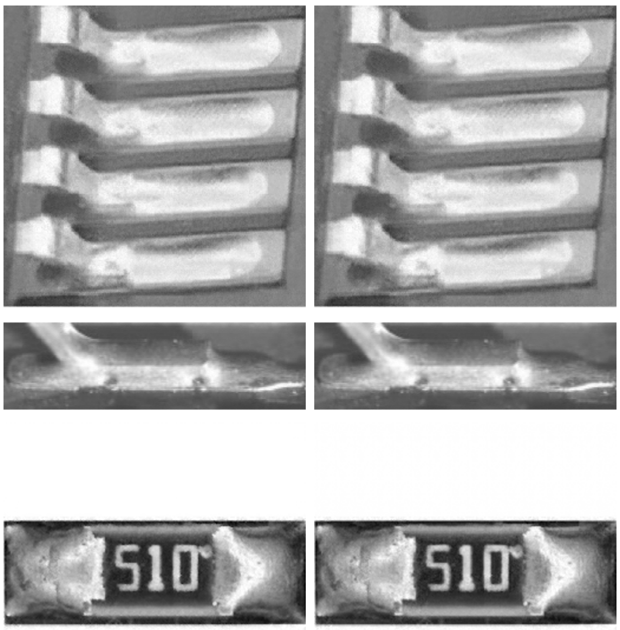
V. Conclusion

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 Γ 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

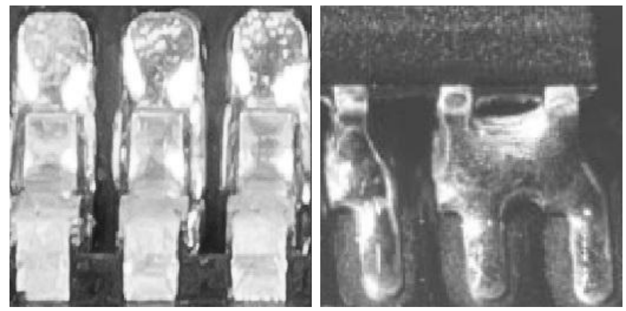
Methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
SP	8.746/0.079	12.599/0.159	15.88/0.249	18.46/0.333	20.55/0.404	22.57/0.477	24.30/0.537	25.64/0.585	26.89/0.631
OMP	13.42/0.166	17.258/0.267	19.76/0.361	21.81/0.439	23.57/0.504	25.14/0.563	26.49/0.612	27.61/0.651	28.58/0.684
GBP	0 /0	14.14/0.239	16.90/0.335	18.87/0.406	20.44/0.460	21.86/0.509	23.21/0.552	24.24/0.585	25.21/0.613
CoSaMP	7.49/0.055	10.89/0.113	14.55/0.199	17.47/0.290	19.95/0.369	22.38/0.457	24.37/0.529	25.94/0.590	27.32/0.645
BOMP	17.81/0.551	20.952/0.646	23.14/0.709	24.95/0.761	26.46/0.802	27.92/0.838	29.21/0.866	30.31/0.887	31.31/0.904
BCoSaMP	8.39/0.067	10.991/0.115	13.62/0.166	16.22/0.234	18.47/0.306	20.80/0.395	23.12/0.480	24.83/0.547	26.25/0.604
FGB	9.56/0.137	17.961/0.476	20.60/0.646	22.42/0.734	23.67/0.782	25.15/0.821	26.46/0.852	27.31/0.876	28.55/0.898
BFGB	13.41/0.166	21.23/0.615	23.02/0.768	24.35/0.801	25.26/0.827	27.40/0.876	28.21/0.891	29.14/0.913	29.87/0.929
BCS_SPL	25.58/0.821	29.512/0.888	31.70/0.915	33.73/0.939	35.65/0.956	37.52/0.969	39.55/0.979	42.19/0.988	45.87/0.994
BCS_WP_SPL	**25.63/0.822**	**29.606/0.890**	**32.33/0.918**	**34.40/0.941**	**36.39/0.959**	**38.29/0.971**	**40.26/0.980**	**43.06/0.990**	**46.73/0.995**



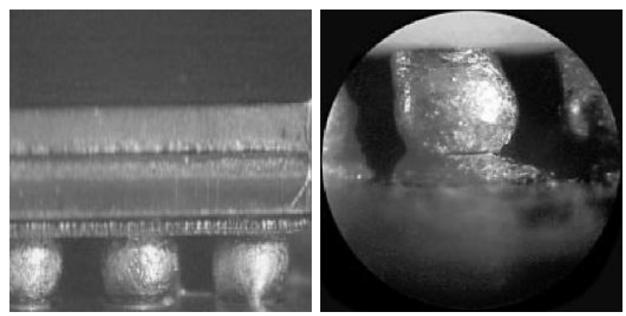
(a) BCS_SPL (b) BCS_WP_SPL

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

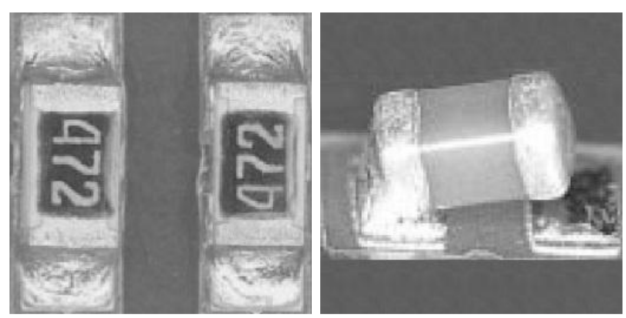
cording to PSNR and SSIM than classical algorithms for reconstruction of images of different types of solder joints.
- With different levels $J = 2,3$, a 3×3 Wiener filter achieves better results according to PSNR and SSIM than 5×5 and 7×7 Wiener filters. When the sampling rate $u < 0.5$ the proposed method based on 2 level wavelet packet decomposition can achieve a 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 can achieve a better results than 2 level wavelet packet in PSNR and SSIM.
- 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.



(a) Gull-wing (acceptance) (b) Gull-wing(defect)



(c) BGA (acceptance) (d) BGA (defect)



(e) Chip-component (acceptance) (f) Chip-component (defect)

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 Technology 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 (Hengkefa [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

[1] Acciani, G., Brunetti, G., Fornarelli, G.: Application of neural networks in optical inspection and classification of solder joints in surface mount technology. IEEE Transactions on industrial informatics **2**(3), 200–209 (2006)

[2] Anselmi, N., Oliveri, G., Hannan, M.A., Salucci, M., Massa, A.: Color compressive sensing imaging of arbitrary-shaped scatterers. IEEE Transactions on Microwave Theory and Techniques **65**(6), 1986–1999 (2017)

[3] Bi, D., Xie, Y., Li, X., Zheng, Y.R.: Efficient 2-d synthetic aperture radar image reconstruction from compressed sampling using a parallel operator splitting structure. Digital Signal Processing **50**, 171–179 (2016)

[4] Chen, Z., Hou, X., Qian, X., Gong, C.: Efficient and robust image coding and transmission based on scrambled block compressive sensing. IEEE Transactions on Multimedia **20**(7), 1610–1621 (2018)

[5] Cui, H., Zhang, S., Gan, X., Shen, M., Wang, X., Tian, X.: Information recovery via block compressed sensing in wireless sensor networks. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6 (2016)

[6] Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing signal reconstruction. Information Theory, IEEE Transactions on **55**(5), 2230–2249 (2009)

[7] Davenport, M.A., Needell, D., Wakin, M.B.: Signal space CoSaMP for sparse recovery with redundant dictionaries. IEEE Transactions on Information Theory **59**(10), 6820–6829 (2013)

[8] Deswal, S., Singhania, S., Gupta, S., Garg, P.: An optimised fuzzy approach to remove mixed noise from images. International Journal of Signal Processing, Image Processing and Pattern Recognition **9**(4), 293–322 (2016)

[9] Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)

[10] Donoho, D.L., Elad, M., Temlyakov, V.N.: Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Transactions on Information Theory **52**(1), 6–18 (2006)

[11] Elad, M., Figueiredo, M.A.T., Yi, M.: On the role of sparse and redundant representations in image processing. Proceedings of the IEEE **98**(6), 972–982 (2010)

[12] Eldar, Y.C., Kuppinger, P., Blcskei, H.: Block-sparse signals: Uncertainty relations and efficient recovery. IEEE Transactions on Signal Processing **58**(6), 3042–3054 (2010)

[13] Gan, L.: Block compressed sensing of natural images. In: 15th International Conference on Digital Signal Processing, pp. 403–406. IEEE (2007)

[14] Haupt, J., Nowak, R.: Signal reconstruction from noisy random projections. IEEE Transactions on Information Theory **52**(9), 4036–4048 (2006)

[15] Hurley, N., Rickard, S.: Comparing measures of sparsity. IEEE Transactions on Information Theory **55**(10), 4723–4741 (2009)

[16] Liu, H., Wang, W.: Block compressed sensing reconstruction with adaptive-thresholding projected landweber for aerial imagery. Journal of Applied Remote Sensing **9**(1), 095037–095037 (2015)

[17] Melli, S.A., Wahid, K.A., Babyn, P., Montgomery, J., Snead, E., El-Gayed, A., Pettitt, M., Wolkowski, B., Wesolowski, M.: A compressed sensing based reconstruction algorithm for synchrotron source propagation-based X-ray phase contrast computed tomography. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **806**, 307–317 (2016)

[18] Mousavi, A., Dasarathy, G., Baraniuk, R.G.: DeepCodec: Adaptive sensing and recovery via deep convolutional neural networks (2017)

[19] Mun, S., Fowler, J.E.: Block compressed sensing of images using directional transforms. In: International Conference on Image Processing, pp. 3021–3024. IEEE (2009)

[20] Niu, M., Salari, S., Chan, F., Rajan, S.: Recovery probability analysis for sparse signals via OMP. IEEE Transactions on Aerospace and Electronic Systems **51**(4), 3475–3479 (2015)

[21] Parikh, N., Ream, J.M., Zhang, H.C., Block, K.T., Chandarana, H., Rosenkrantz, A.B.: Performance of simultaneous high temporal resolution quantitative perfusion imaging of bladder tumors and conventional multi-phase urography using a novel free-breathing continuously acquired radial compressed-sensing MRI sequence. Magnetic Resonance Imaging **34**(5), 694–698 (2016)

[22] Sun, Y., Gu, F.: Compressive sensing of piezoelectric sensor response signal for phased array structural health monitoring. International Journal of Sensor Networks **23**(4), 258–264 (2017)

[23] Tsilifis, P., Huan, X., Safta, C., Sargsyan, K., Lacaze, G.: Compressive sensing adaptation for polynomial chaos expansions. Journal of Computational Physics **380**, 29 – 47 (2019)

[24] Unde, A.S., Deepthi, P.: Block compressive sensing: Individual and joint reconstruction of correlated images. Journal of Visual Communication and Image Representation **44**, 187–197 (2017)

[25] Van Den Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. SIAM Journal on Scientific Computing **31**(2), 890–912 (2008)

[26] Wang, F., Li, J., Liu, S., Zhao, X., Zhang, D., Tian, Y.: An improved adaptive genetic algorithm for image segmentation and vision alignment used in microelectronic bonding. IEEE/ASME Transactions On Mechatronics **19**(3) (2014)

[27] Wang, S., Su, Z., Ying, L., Xi, P., Zhu, S., Feng, L., Feng, D., Dong, L.: Accelerating magnetic resonance imaging via deep learning. In: IEEE International Symposium on Biomedical Imaging (2016)

[28] Xin, Z., Li, D.: Improvement of Gauss random measurement matrix. Foreign Electronic Measurement Technology (2017)

[29] Yan, Y., Jian, S., Li, H., Xu, Z.: ADMM-Net: A deep learning approach for compressive sensing MRI (2017)

[30] Yao, H., Feng, D., Zhang, D., Ma, Y., Zhang, S., Zhang, Y.: DR^2-Net: Deep residual reconstruction network for image compressive sensing (2017)

[31] Ye, Q., Cai, N., Li, J., Li, F., Wang, H., Chen, X.: IC solder joint inspection based on an adaptive-template method. IEEE Transactions on Components, Packaging and Manufacturing Technology **8**(6), 1121–1127 (2018)

[32] You, H., Zhu, J.: Image reconstruction based on block-based compressive sensing. In: Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015), vol. 27, p. 30 (2015)

[33] Zhang, Y., Zhang, L.Y., Zhou, J., Liu, L., Fei, C., Xing, H.: A review of compressive sensing in information security field. IEEE Access **4**, 2507–2519 (2017)

[34] Zhao, C., Ma, S., Zhang, J., Xiong, R., Gao, W.: Video compressive sensing reconstruction via reweighted residual sparsity. IEEE Transactions on Circuits and Systems for Video Technology **27**(6), 1182–1195 (2017)

[35] Zhao, H., Wang, Y., Qiao, Z., Fu, B.: Solder joint imagery compressing and recovery based on compressive sensing. Soldering & Surface Mount Technology **26**(3), 129–138 (2014)

[36] Zhao, H., Zhao, H., Chen, J., Chen, J., Xu, S., Xu, S., Wang, Y., Wang, Y., Qiao, Z., Qiao, Z.: Compressive sensing for noisy solder joint imagery based on convex optimization. Soldering & Surface Mount Technology **28**(2), 114–122 (2016)

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.