A FAST SPARSE RECONSTRUCTION APPROACH FOR HIGH DIMENSIONAL IMAGE-BASED OBJECT SURFACE ANOMALY DETECTION

Woon Huei Chai, Chi-Keong Goh†, Shen-Shyang Ho*, and Hiok Chai Quek*

Rolls-Royce @ Nanyang Technological University Corporate Lab, Singapore
Energy Research Institute, Interdisciplinary Graduate School, Nanyang Technological University, Singapore
* School of Computer Engineering, Nanyang Technological University, Singapore
† Rolls-Royce Singapore Pte Ltd, Singapore

ABSTRACT

We propose an approach to resolve two issues in a recent proposed sparse reconstruction based anomaly detection approach as a part of automated visual inspection (AVI). The original approach needs large computation and memory for high-dimensional problem. To solve it, we proposed a two-step sparse reconstruction, 1) the first sparse representation of input image is estimated in a sparse reconstruction with low dimensional downsampled images and 2) the high dimensional residual values is generated in another sparse reconstruction with the sparse representation. The first step provides the flexibility of freely adjusting the computation and the demand of memory storage with small trade-off of detection accuracy. Moreover, an illumination adaptive threshold with morphological operators is used in the anomaly classification. Empirical results show that the proposed approach can effectively replace the original approach with better results.

Index Terms— automated visual inspection, anomaly detection, sparsity, surface defect, adaptive thresholding

1. INTRODUCTION

Anomaly detection is important and widely used in various domain to detect patterns which do not conform normal patterns. Its application includes intrusion detection [1], fraud detection [2], electronics fault diagnosis [3] terror-related activities [4], visual inspection [5], etc.

Visual inspection is a process to detect defects which affect quality of product. It is difficult to develop a robust AVI technique which can tackle all the common industrial inspection challenges such as 1) small anomalous region embedded within majority normal background, 2) characteristics within normal regions are diverse, 3) characteristics within anomalies are diverse, 4) small availability of anomalous samples, 5) the inspected data has extreme high dimensionality.

Sparse representation is used to represent and reconstruct signals and is widely used in image-based problem. The recently proposed sparse reconstruction approach [5] for defect detection is severely limited by its large computation and memory requirements.

This paper presents a fast sparse reconstruction approach to overcome the above mentioned limitations. The details of the proposed approach is elaborated in the next section. Section 3 shows a two-step optimization algorithm for learning sub-optimal task dependent parameters from a global convex and local non-convex optimization problem. Section 4 shows the empirical results including the comparisons of anomaly detection results of both new proposed and original sparse reconstruction approaches [5]. Section 5 concludes this paper.

2. METHODOLOGY

Figure 1 illustrates the flow chart describing the methodology of our proposed approach. The flow chart of our proposed anomaly detection approach is in Figure 1. The novelty of the new proposed fast sparse reconstruction is that it is capable to detect anomalies on high resolution images which the original approach [5] is limited to. It provides the flexibility of freely adjusting the computation and the demand of memory storage with small trade-off of detection accuracy. Moreover, a novel threshold which adapts the illumination condition in input image replaces the original universal threshold.

2.1. Fast Sparse Reconstruction

Both original \( w' \times h' \)-dimensional input image \( I' \) (\( w' \) and \( h' \) are width and height of original input image) and its \( w \times h \)-dimensional downsampled image \( I \) are firstly concatenated either row-wise or column-wise to form \( p' \) (\( p' = w' \times h' \)) and \( p \) (\( p = w \times h \))-dimensional vectors, \( x' \) and \( x \). The \( p \)-dimensional input data vector, \( x = (x_1, x_2, \ldots, x_p)^T \), can be reconstructed with a \( p \times m \)-dimensional matrix, \( D \) (dictionary/design matrix with \( m \) words/atoms), and a \( m \)-dimensional vector, \( \alpha \), in the equalization form of \( x = D\alpha + \lambda e \), where \( \lambda \) is a constant and \( \lambda e \) is a \( p \)-dimensional residual vector caused by noises or anomaly. The \( \alpha \) and the equalization are called sparse representation of image \( I' \) and \( I \) and sparse reconstruction if \( \alpha \) is a sparse vector. The values of \( \alpha \) and \( e \) can be found through the following optimization problem:

\[
\min_{\alpha, e} \|\alpha\|_1 + \|e\|_1 \quad (1)
\]

s.t. \( x = D\alpha + \lambda e \) where \( \lambda \) is a parameter which controls the trade-off between sparsities of \( \alpha \) and \( e \). \( \|\alpha\|_1 \) ensures only few relevant atoms are chosen in the reconstruction and \( \|e\|_1 \) is used due to the
prior knowledge that anomaly is only a small portion in $x$ (most values in residual vector should be around zero).

In our problem, $D$ stores the global information of normal samples. Each atom of $D$ is a $p$-dimensional vectorized normalized downsampled anomaly free image. In general, $D$ can be constructed with dictionary learning [6, 7] too. After the sparse representation $\alpha$ is estimated, the sparse reconstruction at high resolution (2) is conducted to have the $p'$-dimensional residual values $e'_{x}$ which can be further reorganized to $w' \times h'$-dimensional residual image at original resolution for anomaly classification, where atoms in $D'$ are vectorized normalized anomaly free images at original resolution with the same order of atoms in $D$.

$$x' = D'\alpha + e'.$$

(2)

Note that formula (1) is the computation and memory storage bottleneck of the whole sparse reconstruction process if $p$ is large. However, we can adjust the value of $p$ freely with any downsampling resolution with a small trade-off in final anomaly detection accuracy to speed up the process and reduce the memory storage requirement for the process. Hence, the whole process is called fast sparse reconstruction.

Also note that it saves the computation for estimating $\alpha$ and high dimensional $e$ using high dimensional (high resolution images) $x$ and $D$ with original approach does in [5].

2.2. Anomaly Classification

Firstly, all the values in $e'$ are rounded to integers between 0 and 255. The vector of rounded $e'$ is called residual vector $e'_{x}$ and the image form of $e'_{x}$ is called preprocessed residual image $R'$. All pixels $x'_{i}$ are classified into two classes, normal and anomalous pixels, based on the classifier $C(x'_{i})$ as follows:

$$C(x'_{i}) = \begin{cases} 1 & \text{if } |e'_{x,i}| > T > 0 \land D_{l} \leq D_{l}(T) \leq D_{u} \\ 0 & \text{otherwise} \end{cases}$$

(3)

where $x'_{i}$ is a pixel in the input image. $C(x'_{i})$ returns 1 (estimated anomalous pixel) if the absolute residual value, $|e'_{x,i}|$, at that pixel is higher than threshold $T$. The pixel is in a connected region which absolute values are higher than threshold and the dimensions $D_{l}(T)$ of the smallest bounding box which bounds the connected region is between the lower and the upper bounds, $D_{l}$ and $D_{u}$. The bounding box constraint is for removing noise and fulfilling the inspection requirements (i.e. the tolerance of acceptable defect and unacceptable anomaly specifications such as sizes of defects). Otherwise, $C(x'_{i})$ returns 0 (estimated normal pixel). Morphological operations (opening and closing morphological operations of square kernel of size $S_{k}$) may be applied on the thresholded residual image before forming the connected region and estimate the labels for removing noise.

Note that, the threshold $T$ in formulation (3) is designed to adapt the illumination condition in the image. It replaces the universal threshold used in the original approach [5].

In (3), the main challenge is the selection of the threshold $T$. Let $N_{c}(t)$ be the number of connected estimated anomalous pixel regions after applying (3) with $T = t$. Define

$$T = \min\{|t|t = 0, 1, \ldots, 255, N_{c}(t) \geq \eta\},$$

(4)

where $t$ is an integer between 0 and 255 and $\eta$ is a pre-defined parameter. It is the lowest integer threshold which has at least $\eta$ connected estimated anomalous pixel regions. $\eta$ is dataset dependent (i.e. the characteristics of noise and the number of anomaly occurrence in an image). Hence, $\eta$ needs to be re-estimated when new dataset has different characteristics (i.e. the changes of target defect type, the changes of equipment of the environment, the image acquisition devices, the inspected product, etc.).

3. PARAMETER OPTIMIZATION

In formulas (1) and (4), there are two dataset dependent parameters, $\lambda$ and $\eta$ need to be learned. These two parameters depend on the characteristics of anomaly to be detected, the environment, the image acquisition devices, the inspected product, etc. The flow chart of our proposed anomaly detection approach.
products, etc. Based on our analysis\(^1\), the optimization problem of accuracy (5) with changing either one of the two parameters is convex globally and non-convex locally respectively. Hence, we proposed a specific supervised parameter learning algorithm to tackle this special non-convex problem.

The parameter learnings are done with a two-step greedy search in Algorithm 1. Algorithm 2 and 3 learn \(\lambda\) and \(\eta\) respectively. The two Algorithms (Step 4 and 5 in Algorithm 1) are iterated until either the validation accuracy converges (depends on the stopping criterion of accuracy, \(S_1\)) or the maximum number of iterations \(S_2\) is reached.

**Algorithm 1:** Finding best \(\lambda\), \(\eta\) and \(T\)

**Input:** \(I_t\), set of labeled training images; \(D, D'\), dictionaries with atoms at low and high resolutions respectively; \(S_1, S_2\), stopping criteria of accuracy and number of iteration; \(N\), search granularity; \(S_k\), kernel size; \(\beta\), trade-off coefficient

**Output:** \(\lambda_0\) and \(\eta_0\)

\[\begin{align*}
1: & \quad \eta_0 = 1000; \lambda_0 = 0; D_1 = 0; D_u = 1000; N_m = 0; \text{maxAccp} = 1; \text{maxAcc} = 0; \eta_{\text{min}} = 0.0001; \lambda_{\text{max}} = 5; \eta_{\text{max}} = 1000 \\
2: & \quad \text{while } \|\text{maxAcc} - \text{maxAccp}\| > S_1 \text{ or } N_m < S_2 \text{ do} \\
3: & \quad N_m = N_m + 1; \lambda_p = \lambda_0; \eta_p = \eta_0; \text{maxAcc} = \text{maxAcc} \\
4: & \quad \lambda_0, \text{maxAcc} = \text{Parameter optimization} (I_t, D, D', S_1, S_2, N, S_k, D_1, D_u, \beta, \lambda_{\text{min}}, \lambda_{\text{max}}, \eta_0, 1) \\
5: & \quad \eta_0, \text{maxAcc} = \text{Parameter optimization} (I_t, D, D', S_1, S_2, N, S_k, D_1, D_u, \beta, \eta_{\text{min}}, \eta_{\text{max}}, \lambda_0, 2) \\
6: & \quad \text{if } \text{maxAcc} > \text{maxAccp} \text{ then} \\
7: & \quad \text{maxAcc} = \text{maxAccp}; \lambda_0 = \lambda_p; \eta_0 = \eta_p \\
8: & \quad \text{break} \\
9: & \quad \text{end if} \\
10: & \quad \text{end while}
\end{align*}\]

In Algorithm 1, Step 1 initializes all the parameters and variables required for the two-step greedy search. \(\eta_0\) and \(\lambda_0\) are the optimized \(\eta\) and \(\lambda\); \(D_1\) and \(D_u\), the lower and upper bounds of the dimensions of the smallest bounding box size which bound a connected region which is being classified as anomaly; \(N_m\) is the iteration number of the main loop; \(\text{maxAccp}\) and \(\text{maxAcc}\) are the validation accuracies with previous and current optimized parameters respectively; \(\lambda_{\text{min}}, \lambda_{\text{max}}, \eta_{\text{min}}\) and \(\eta_{\text{max}}\) are the lower and upper bounds of \(\lambda\) and \(\eta\) for the parameter searches. Steps 2 to 10 is the main optimization loop. The whole loop ends when either the difference between the previous and the latest optimized accuracies is not longer larger than \(S_1\) or the iteration number is equal to \(S_2\). Step 3 calculates the iteration number \(N_m\) of the main loop and update the most recent optimized parameters and theirs corresponding accuracies. Steps 4 and 5 optimize \(\lambda\) and \(\eta\) respectively by fixing the other parameter. Step 6 to 9 check whether the current optimized accuracy is lower than the previous one. The loop is broken if it is lower and the optimized parameters are set to be equal to the previous one. Otherwise the loop continues with the stopping criteria stated at Step 2.

In Algorithm 2, the input \(P_c\) (1 or 2) states the parameter \((\lambda\) or \(\eta\)) to be learnt. The specified parameter is learnt and is returned with its corresponding anomaly detection accuracy. Step 1 initializes the iteration number \(N\) of the optimization loop (Step 2 to 16) and the variables \(\text{acc}_a\) and \(\text{acc}_b\) and \(\text{acb}\) are the highest two accuracies determined from the parameter search list. The optimization loop stops when either one of the stopping criteria, \(S_1\) or \(S_2\) is met. Step 3 calculates the iteration number \(N\). Step 4 to 14 compute all accuracies with different parameter values defined at Step 5. The values depend on the search granularity \(N\), the lower and upper bounds, \(P_{\text{min}}\) and \(P_{\text{max}}\) of parameter search range. The accuracy of each training image is computed with Algorithm 3 at either Step 8 or 10 depending on which parameter is learnt. Step 15 updates the lower and upper bounds of the search range for next iteration.

Algorithm 3 computes the weighted anomaly detection accuracy with formula (5).

\[
\text{Acc}_o(\beta) = \beta (\text{specificity}) + (1 - \beta) (\text{specificity}) = \beta \frac{TP}{P} + (1 - \beta) \frac{TN}{N},
\]

where \(TP, TN, P\) and \(N\) are the number of true positives, true negatives, positives and negatives respectively. \(\text{Acc}_o\) is the sum of the weighted sensitivity (recall) [8] and specificity [8] measures. \(\text{Acc}_o\) alleviates the imbalanced dataset challenges in many real-world problems especially for anomaly detection task which has infrequent anomalies.

**Algorithm 2:** Parameter optimization

**Input:** \(I_t; D, D'; S_1; S_2; N; S_k; D_1; D_u; \beta; P_{\text{min}}; P_{\text{max}}\), lower and upper bound of parameter search range; \(P_{c}\), fixed parameter; \(P_c\), parameter choice \(1\) for \(\lambda\), \(2\) for \(\eta\)

**Output:** \(P_o\) and \(\text{Acc}_o\)

\[\begin{align*}
1: & \quad N_v = 0; \text{acc}_a = 1; \text{acc}_b = 0 \\
2: & \quad \text{while } \|\text{acc}_a - \text{acc}_b\| > S_1 \text{ or } N_v < S_2 \text{ do} \\
3: & \quad N_v = N_v + 1 \\
4: & \quad \text{for } i = 0 \text{ to } N \text{ do} \\
5: & \quad \text{P}\_i = \text{P}\_\text{min} + \frac{i}{N} (\text{P}\_\text{max} - \text{P}\_\text{min}) \\
6: & \quad \text{for } j = 1 \text{ to } N_t \text{ do} \\
7: & \quad \text{if } P_c = 1 \text{ then} \\
8: & \quad \text{acc}\_i[j] = \text{Accuracy computation} (I_t[j], D, D', P_i, P_o, S_k, D_1, D_u, \beta) \\
9: & \quad \text{else if } P_c = 2 \text{ then} \\
10: & \quad \text{acc}\_i[j] = \text{Accuracy computation} (I_t[j], D, D', P_i, P_o, S_k, D_1, D_u, \beta) \\
11: & \quad \text{end if} \\
12: & \quad \text{end for} \\
13: & \quad \text{acc}\_i = \sum_{j=1}^{N_t} \frac{1}{N_t \text{ } N_t} \text{ acc}\_i[j] \\
14: & \quad \text{end for} \\
15: & \quad k_1 = \text{arg max}_{k_1} (\text{acc}\_i); k_2 = \text{arg max}_{k_2} (\text{acc}\_i \setminus \text{acc}\_i[k_1]); \text{acc}\_i = \text{acc}\_i[k_1]; \text{acb} = \text{acc}\_i[k_2]; P_o = P[k_1]; P_{\text{min}} = \min(P[k_1], P[k_2]); P_{\text{max}} = \max(P[k_1], P[k_2]) \\
16: & \quad \text{end while}
\end{align*}\]

17: \(\text{Acc}_o = \text{acc}_i[k_1]

18: \text{Return } P_o \text{ and } \text{Acc}_o

\(^1\)Due to limitation of article’s length, the details of the analysis will be presented in a journal which we are going to submit soon.
4. EXPERIMENTAL RESULTS

4.1. Metallic Object Dataset

The dataset is a subset of a large number of images collected from an AVI process for quality control in a metallic object manufacturing. Anomalies are the defects on metallic surfaces. The dataset consists of 891 images with original resolution of 2050 × 2448. There are three defect types: “Melt”, “Plus Metal”, and “Scuff”. Each 2 instances of same defect type are grouped together for an experimental trial. There are 7 instances: “Melt A” (58 [images contain defects], 69 [defect free images]), “Melt B” (58, 161), “Melt C” (58, 69), “Plus Metal A” (53, 69), “Plus Metal B” (52, 23), “Scuff A” (53, 46) and “Scuff B” (57, 63). All instance pairs are organized as follows: “Melt 1” (“Melt A” and “Melt B”), “Melt 2” (“Melt A” and “Melt C”), “Melt 3” (“Melt B” and “Melt C”), “Plus Metal” (“Plus Metal A” and “Plus Metal B”) and “Scuff” (“Scuff A” and “Scuff B”). All images in the same instance are taken from the same viewpoint with different illumination conditions. In each experimental trial, each instance takes turn to be training and testing sets.

4.2. Comparisons of Proposed and Original Approaches

The default values for parameter and variable initialization are declared in all algorithms presented in Section 2. Figure 2 shows four examples of anomaly detection results based on our proposed approach. Balanced accuracy (weighted accuracy in (5) with $\beta = 0.5$) is used as performance measure in all experiments described in this subsection. Figure 3 shows the results of all experimental trials. The new proposed approach outperforms the original approach in low resolution output. The performance of both approaches converges in high resolution output.

Note that our proposed approach spends average of 0.0446 seconds to solve the optimization problem (1) for one image at resolution of $25 \times 25$ using linear programming and “Scuff” defects. Blue and red boxes are the smallest bounding boxes which bound individual connected True Positive and False Positive respectively.

<table>
<thead>
<tr>
<th>Algorithm 3: Accuracy Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $I$, input image; $D_1$, $D_2$; $\lambda$; $\eta$; $S_k$; $D_h$; $\beta$</td>
</tr>
<tr>
<td><strong>Output:</strong> Accuracy</td>
</tr>
<tr>
<td>1: Find sparse representation, $\alpha$, using (1) for $I$ with $D$ and $\lambda$</td>
</tr>
<tr>
<td>2: Find residual vector, $e^\prime$, using (2) with $D'$ and $\alpha$</td>
</tr>
<tr>
<td>3: Round all the values in the residual vector to integers between 0 and 255 and to have the residual image $R'$</td>
</tr>
<tr>
<td>4: Find the threshold $T$ using (4) with given $S_k$ and $\eta$</td>
</tr>
<tr>
<td>5: Apply (3) to obtain the labels of all pixels with $T$, $D_1$, $D_h$ and $S_k$</td>
</tr>
<tr>
<td>6: Calculate Accuracy with $\beta$ and (5)</td>
</tr>
<tr>
<td>7: Return Accuracy</td>
</tr>
</tbody>
</table>

---

2Due to confidentiality, the description is limited.
7. REFERENCES


