






A-BEBLID: A Hybrid Image Registration Method for Lithium-ion Battery Cover Screen Printing

Na Wang , *Member, IEEE*, Ying Chen , Xianyong Zhang* , Xuhong Zhang , and Raymond Chiong , *Senior Member, IEEE*

Abstract—To address the problem of miss- and false-detection during quality inspection of lithium-ion battery cover screen printing (LBCSP), we propose a hybrid image registration method using a point-based feature extraction algorithm and nonlinear-scale space construction. Our proposed method integrates the AKAZE algorithm with the BEBLID descriptor, and is therefore named A-BEBLID. Facing the challenge of the inevitable offset caused by machine vibration during production, we combine a nonlinear diffusion filter with a local image descriptor to extract features from images, and then use the GMS algorithm to remove the wrong matching pairs. We tested the method on a dataset we created using images taken from actual lithium-ion battery production lines, named LBCSP. We also evaluated the method on the public HPatches dataset. The average precision achieved by A-BEBLID on the LBCSP dataset is 89% (threshold: 2 pixels), with a localization error of 1.11 pixels, while on the HPatches dataset, the average precision is 73% (threshold: 2 pixels), with a localization error of 1.52 pixels. Comprehensive experimental results also showed that the proposed A-BEBLID can outperform other approaches being compared to. The method can be further applied to other industry scenarios with similar image registration requirements.

Index Terms—A-BEBLID, AKAZE, BEBLID, GMS, image registration, lithium-ion battery cover screen printing, point-based features

I. INTRODUCTION

SCREEN printing methods are commonly used for the printing of lithium-ion battery product covers. As part of quality control, manufacturers need to monitor the quality of screen printing on the production lines in real-time. During production, a slight vibration may cause the printed image to

rotate and/or shift, making it more difficult for the computer to locate the desired features, such as the ink blocking area, resulting in miss- and false- detection.

Image registration is a technique of matching two or more images of the same object taken in different environments, i.e., from different angles, by different cameras, at different times, etc [1]. The task is to identify and align a big amount of visual information from different sensors, and produce more visual representations for downstream tasks [2]. It can be applied to image enhancing, image mosaic, object tracking, image segmentation, etc. Numerous studies have been conducted in the area of image registration, and the approaches include intensity-based, feature-based, and deep learning-based methods. Although deep learning has been applied to various computer vision tasks [3], due to its poor interpretability and high computational cost, it is a less-desired approach for lithium-ion battery cover screen printing (LBCSP).

The intensity-based methods use a series of transformation matrices to align two images by warping the tested image to the coordinates of the reference image [4]. Feature-based methods, on the other hand, search for specific information such as points, lines, contours and polygon structures that would serve as key factors in determining the similarity of the compared images. Point-based features, easy to obtain and less affected by noise [5], provide an effective solution for applications such as the screen printing of lithium-ion battery covers. Our work here therefore focuses on the feature-based methods.

In 1977, Moravec [6] proposed a method for point-based feature extraction, and his method has piqued the interest of researchers, academics, and industry leaders since then. Harris et al. [7] optimized the method using a local auto-correlation function and demonstrated robust performance against rotation change and noise interference. While human eyes can identify an object regardless of its size, computers have been struggling to achieve the same ability. In 1999, Lowe [8] made a breakthrough when he proposed the object recognition system. In 2004, he improved the system and renamed it scale-invariant feature transform (SIFT) [9]. In 2008, Bay et al. [10] proposed the speeded up robust features (SURF) method, which improved the algorithm with higher execution efficiency using a Hessian matrix-based detector and a distribution-based descriptor. Rublee et al. proposed a binary descriptor based on features based on the accelerated segment test (FAST) [11] and binary robust independent elementary

Submitted on 24-May-2022. This work was supported in part by the Guangzhou Science and Technology Planning Project Grand (201902020003), in part by the Guangzhou Key Laboratory of Intelligent Building Equipment Information Integration and Control (202002010003), and in part by Special Project for Key Research Fields of Colleges and Universities in Guangdong Province 2021 (New Generation Information Technology) (2021zdzx1033).

Na Wang (wangna@gpnu.edu.cn), Xianyong Zhang (zhangfriend-jun@163.com) and Xuhong Zhang (zhxhzhy@163.com) are with the School of Automation, Guangdong Polytechnic Normal University, 293 Zhongshan Avenue West, Tianhe District, Guangzhou 510665, China.

Ying Chen is with the Department of Intelligent Engineering, Guangdong Maoming Agriculture & Forestry Technical College, Guangdong 525099, China (e-mail: cying0012@163.com).

Raymond Chiong is with the School of Information and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: raymond.chiong@newcastle.edu.au).

features (BRIEF) [12]. The method was named ORB [13].

Alcantarilla et al. proposed the KAZE algorithm (KAZE means wind in Japanese) that detects and describes image features through a nonlinear diffusion filter [14]. To speed up the method, its improved version, accelerated-KAZE (AKAZE), embedded a fast explicit diffusion (FED) in the framework [15]. Their experimental results showed that AKAZE was faster not only than KAZE, but also SURF and SIFT. In 2020, Suárez et al. adopted AdaBoost with a weak-learner training scheme to generate local descriptions, and they proposed the boosted efficient binary local image descriptor (BEBLID). Their experimental results demonstrated that the BEBLID has higher computing efficiency than ORB and an accuracy close to SIFT [16].

In this paper, we propose an A-BEBLID framework that combines the AKAZE algorithm and the BEBLID descriptor for image registration. To increase the robustness against rotation and position shifting, we first use the AKAZE algorithm to perform nonlinear diffusion filtering, and thus construct a nonlinear-scale space. Then, we adopt the BEBLID descriptor to improve efficiency. Afterward, we remove the incorrect matching pairs using the Grid-based Motion Statistics (GMS) algorithm [17]. Our proposed method was tested on the HPatches public dataset [18] as well as a dataset we created using images taken from actual LBCSP production lines. This LBCSP dataset consists of 5 folders: scale, rotation, shift, affine, and features. The features were saved as .txt and .csv files of features extracted using six different methods: SIFT, SURF, ORB, AKAZE, ORB+BEBLID, and A-BEBLID. An overview of the dataset structure is shown in Fig. 1. Experimental results confirmed that the proposed A-BEBLID can outperform other methods under comparison.

The rest of this paper is organized as follows. Section II reviews related work on image registration methods. Section III describes the AKAZE algorithm and the BEBLID descriptor, as well as the framework of the proposed method. Section IV presents the experiment setup and evaluation indexes. Section V discusses the experimental results and findings. Finally, conclusions are drawn in Section VI.

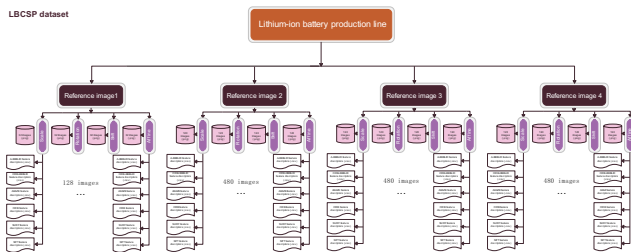


Fig. 1: Overview of the LBCSP dataset. There are 1568 images in the dataset plus four template images. For the first template, there are 32 images classified into four groups: scale, rotation, shift, and affine, a total of 128 images. For templates 2 – 4, each has 120 images classified into four groups.

II. RELATED WORK

Image registration is a notable research area in computer vision that can be employed in various automatic image analysis tasks [19]. Many attempts have been made by researchers in the field. Achieving an application-suitable balance between accuracy and efficiency has been challenging [20]. In this section, we briefly review previous work related to the applications of printing defect inspection, for which image registration is a critical step.

Chen et al. developed a printed image registration approach that can find feature regions in shapes like quasi-rectangular and oval. This approach, however, can only be used on perfect shapes [21]. Zhou et al. adopted an image registration technique to inspect geometric and color flaws in a can printing system. Geometric flaws were detected using the structural similarity index measure (SSIM) technique, whereas color defects were determined by the CIEDE2000 color formula. The approach is capable of detecting scratches, stains, blur, lines and color variation faults. Running speed, however, was not considered in the study [22]. Hu et al. used a multi-class support vector machine for printing defect inspection. The vector, which consists of geometric defect features and form defect features, was used as a descriptor. However, nonlinear-scale space was excluded from the study [23]. Liu et al. proposed a multi-edge feature fusion approach to detect faults in sheet-fed printing systems. They created a dataset with more translation, deformation, and uneven illumination variations for the research. For registration, they used SIFT and multi-edge feature fusion methods, yielding a mean average precision (mAP) of 92.65% and a recall of 96.29%. However, the computation speed was not mentioned in the study [24]. Dong et al. used the normalized cross-correlation (NCC) function to discover edge and roundness incompleteness defects for screen printing image defect detection, and it achieved an accuracy of 94.6% [25].

This study adds to the existing literature by:

- 1) creating the LBCSP dataset using images taken from actual lithium-ion battery production lines;
- 2) implementing the BEBLID descriptor on the AKAZE algorithm to obtain a higher speed;
- 3) evaluating the performance of six methods, namely SIFT, SURF, ORB, AKAZE, ORB+BEBLID, and A-BEBLID on the LBCSP and HPatches datasets¹;
- 4) testing the methods on images from actual industry production lines, thereby offering a feasible inspection approach for quality check of similar industrial scenarios.

III. ALGORITHMS

Here, we first introduce the AKAZE algorithm and the BEBLID descriptor, then describe the framework of the proposed method.

A. AKAZE

The AKAZE algorithm was proposed in 2011 by Alcantarilla et al. This method uses a nonlinear diffusion approach

¹<https://github.com/A-BEBLID/LBCSP>

to track the luminance changing of the image. The algorithm is usually described by nonlinear partial differential equations and uses the scale parameter as the divergence factor to control the luminance diffusion of the image, as shown in (1):

$$\frac{\partial H}{\partial t} = \text{Div}(c(x, y, t) \cdot \nabla H), \quad (1)$$

where Div denotes the divergence operation, ∇ represents the gradient, c is the conductivity function, and t is the scale parameter. By specifying a suitable conductivity function, we can make the diffusion adaptive to the local structure.

In 1990, Perona and Malik introduced nonlinear diffusion filtering into the field of computer vision [26]. They made the function c dependent on the gradient magnitude, to reduce the diffusion near edges and to avoid smoothing across borders. The definition of the function c is shown in (2):

$$c(x, y, t) = g(|\nabla H_\sigma(x, y, t)|), \quad (2)$$

where ∇H_σ denotes the gradient of the Gaussian filtered image. The conduction kernel function g is given by (3):

$$g = \frac{1}{1 + |\nabla H_\sigma|^2 / K^2}, \quad (3)$$

where K is the contrast index that changes the level of diffusion.

The nonlinear-scale space of image H can be obtained through (4):

$$H^{i+1, j+1} = (I + \tau_j A(H^i)) H^{i+1, j}, \quad j = 0, 1, 2, \dots, n-1, \quad (4)$$

where I is the identity matrix, and $A(H^i)$ is the conductivity matrix of the image H^i .

AKAZE algorithm uses the fast explicit diffusion (FED) scheme for faster building by varying the step size τ_j , calculated by (5):

$$\tau_j = \frac{\tau_{max}}{2 \cos^2(\pi(2j+1)/(4n+2))}, \quad (5)$$

where τ_{max} is the maximum step when diffusion reaches stability. Similar to SIFT, the scale space in the AKAZE algorithm is discretized in logarithmic steps, in a series of O octaves and S sub-levels, each marked by a discrete octave index o , and a sub-level index s , as shown in (6):

$$\sigma_i(o, s) = \sigma_0 2^{(o + \frac{s}{S})},$$

$$o \in [0, 1, \dots, O-1], i \in [0, 1, \dots, N], s \in [0, 1, \dots, S-1], \quad (6)$$

where i , o , and s represent the index of the image number, octave, and sub-level respectively, σ_0 denotes the base level, and N is the total number of filtered images.

The relation function of the scale parameters σ_i and the diffusion time t_i is shown in (7):

$$t_i = \frac{\sigma_i^2}{2}. \quad (7)$$

The feature detection process begins with a search for the local maximum value of the Hessian determinant after

the normalization of different scales. The calculation of the Hessian matrix is shown in (8):

$$D_{Hessian} = \sigma^2 (D_{xx} D_{yy} - D_{xy} D_{xy}), \quad (8)$$

where D_{xx} and D_{yy} are the second order derivative of the horizontal and vertical directions respectively, and D_{xy} is the second order cross derivative.

The maxima of the level is the point whose determinant is higher than all of its neighbors.

B. BEBLID

In 2020, Suárez et al. introduced BEBLID based on their previous descriptor BELID, in which the BoostedSSC [27] was adopted to select features from descriptions. AdaBoost was implemented to train the descriptor with an unbalanced dataset from the Liberty statue patches to address the heavily asymmetric image matching issue. To reduce similarity loss, all weak learners are given the same weight [16].

Let (x_i, y_i) be the i -th pair of image patches, $l_i = 1$ indicates that both patches have the same salient image structure, whereas $l_i = -1$ indicates they don't. Subsequently, a training dataset can be described as $\{x_i, y_i, l_i\}_{i=1}^N$.

The loss function is defined in (9):

$$\Gamma_{BEBLID} = \sum_{i=1}^N \exp(-\gamma l_i \sum_{k=1}^K h_k(x_i) h_k(y_i)), \quad (9)$$

where γ is the learning rate that determines the training speed and the number of selected weak learners, h_k denotes the k -th weak learner that depends on a feature extraction function: $f(x; T)$.

With a given f and T , the weak learner can then be defined as:

$$h(x) = \begin{cases} +1 & f(x) \leq T \\ -1 & f(x) > T \end{cases}. \quad (10)$$

The image patch can be described by $D(x)$ based on (10). To normalize the output, the output -1 and $+1$ are converted to 0 and 1 respectively, as shown in Fig. 2.

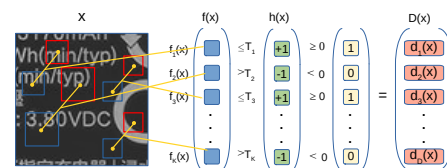


Fig. 2: BEBLID descriptor extraction workflow.

The definition of $f(x)$ is given by (11):

$$f(x; p_1, p_2, s) = \frac{1}{s^2} \left[\sum_{q \in R(p_1, s)} I(q) - \sum_{r \in R(p_2, s)} I(r) \right], \quad (11)$$

where $I(q)$ and $I(r)$ are the grey value of the pixel q and pixel r respectively. The $R(p, s)$ is the square box centered at

pixel p with size s . Therefore, $f(x)$ is the difference between the mean grey values of the pixels in the blue and red boxes in Fig. 2, which are represented by $R(p_1, s)$ and $R(p_2, s)$ in (11) respectively.

The algorithm takes the outputs of $f(x)$ at each pair of patches as input, then finds the threshold that minimizes the weighted classification error.

C. The framework of the proposed method

The method takes advantage of the non-linear space created by the AKAZE algorithm in the initial feature detection step, the features of both images, i.e., the template image and the production line image are located through extreme point detection. These features are then fed to the BEBLID for description using the binary local image descriptor, as shown in Fig. 3.

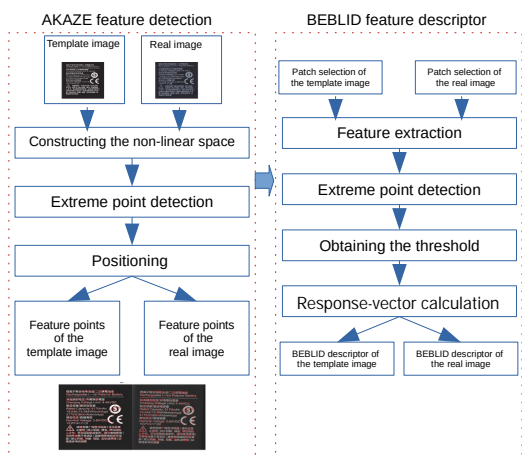


Fig. 3: The framework of A-BEBLID.

IV. EXPERIMENT SETUPS AND EVALUATION INDEXES

Here, we compare the performance of the proposed method with five other methods by testing them on the LBCSP and Hpatches datasets. We first compare the feature point repeatability of the six methods, and then the matching precision, runtime and localization error of these six methods combined with GMS. At last, we input all the above results to the weighted entropy evaluation index (WEEI) formula to evaluate the overall performance.

A. Experimental settings and datasets

The computation environment involved an Intel(R) Core(TM) i5-12600KF CPU with 16G memory and 64-bit Windows11 OS; coding was done using Python 3.8 in Pycharm development tools and the Anaconda integration toolkit. Fig. 4 shows a set of LBCSP images: template, scale, rotation, shift, and affine, where (a) is a template image converted from a standard PDF file. The dataset contains four groups of images to simulate four types of variation in real-world scenarios.

The proposed method is the result of our effort in finding an effective way for the LBCSP image registration. To that end,

when testing the method on the public dataset Hpatches, we chose the ones that are mostly similar to our created dataset in terms of variation. Given that the LBCSP images are taken in a manufacturing workshop with a stable lighting environment, we chose images with viewing angle changes, specifically: the number 2 – 5 images of v_apprentices, v_azzola, v_bees, v_bird, v_circus, v_courses, v_sunseason, and v_vitro, totally 32 to test our method on. This is because the angle changes lead to rotation, shifting, and size change; these are the variations we need to consider in the production lines of this research.



Fig. 4: A set of LBCSP images: (a) template image; (b) scale, the scale range is within [0.5, 1.5] times of the template image; (c) rotation, the rotation range is within [-5°, 5°]; (d) shift, the shift range is within [0.5, 3] pixels; (e) affine, images contain 2 or 3 of the above stated changes.

B. Evaluation metrics

We evaluate the performance using four metrics: runtime, repeatability, matching precision, localization error, and a comprehensive index WEEI.

1) Runtime, given that the quality detection of lithium-ion battery screen printing cover is conducted on the production lines in real-time, it is necessary to test the computing speed of the algorithm. Therefore, runtime is taken as a performance index in this study.

2) Repeatability shows the geometrical stability of the feature points between two images of the same object taken under different conditions [28]. The formula of repeatability rate is shown in (12):

$$R = \frac{\text{Count}[Dis(m_p, n_p)]_{<\epsilon}}{\text{Count}[\min(m_p, n_p)]}, \quad (12)$$

where m_p and n_p are the feature point sets of the template image and the tested real image, respectively. $Dis()$ defines the set of the point pairs (m_p, n_p) whose Euclidean distance is less than the threshold ϵ , and here, ϵ is set to 1.5.

3) Matching precision is defined as the ratio of the number of correct matching pairs to the number of all matching pairs. The formula of matching precision is shown in (13):

$$P = \frac{N_c}{N_r}, \quad (13)$$

where N_c is the amount of correctly matched point pairs of the two images, and N_r is the amount of all matched pairs. **The matching precision of a single image is of little reference significance. Therefore, we calculate the mean value of the precision of all the images in the dataset for evaluation.**

4) Localization error is a statistical result, it reflects the difference between the feature point coordinates of the template image and the tested image. The formula for localization error is shown in (14):

$$E_l = \sqrt{\frac{1}{N_c} \sum_{i=1}^{N_c} [(x_o^i - x_t^i)^2 + (y_o^i - y_t^i)^2]}, \quad (14)$$

where (x_o^i, y_o^i) and (x_t^i, y_t^i) are the reference coordinates and transformed coordinates, respectively.

5) WEEI is the weighted entropy evaluation index, used here for an objective overall evaluation of the six methods. In different application scenarios, the weights should be different depending on the application needs. However, to demonstrate the generalization potential of a method and prevent human judgement from influencing the weight of indexes, WEEI is an objective method when deciding the index weights [29]. The greater the dispersion degree of the index, the greater the impact (i.e. weight) of the index on the overall evaluation. The extreme scenario is where all of an index's values tend to be highly similar or even equal, the index would be of little or no help for assessment. The weight of each index can be calculated using information entropy [30]. The six (each marked as the i -th) methods, are taken as samples to determine the weight vector of the four (each marked as the j -th) indexes. The evaluation index matrix is defined as $X_{6 \times 4} = [x_{ij}]$, where x_{ij} denotes the value of the j -th index of the i -th method.

Next, the matrix $X_{6 \times 4}$ is normalized using the following formula:

$$y_{ij} = \frac{\max(x_j) - x_{ij}}{\max(x_j) - \min(x_j)}, \quad (15)$$

where y_{ij} is the normalized value, $\max(x_j)$ and $\min(x_j)$ are the highest and lowest values of the j -th index among the six methods, respectively.

This formula applies to the indexes whose lower values indicate better performance. Indexes such as repeatability and matching precision, whose higher values indicate better performance are pre-processed by the formula $x_{ij} = 1 - x_{ij}$ before being fed to (15).

With the j -th index, the proportion of the i -th method is defined as p_{ij} , as shown in (16):

$$p_{ij} = y_{ij} / \sum_{i=1}^6 y_{ij}. \quad (16)$$

Next, we calculate the entropy of the j -th index using (17):

$$e_j = \frac{-1}{\ln 6} \sum_{i=1}^6 p_{ij} \ln p_{ij}, \quad (17)$$

where e_j is the entropy of the j -th index, whose lower value indicates a higher amount of information. For this reason, we introduce $d_j = 1 - e_j$, whose higher value indicates a higher amount of information.

Finally, the weight of each index is calculated by (18).

$$w_j = d_j / \sum_{i=1}^4 d_j, \quad (18)$$

where w_j is the weight of each index.

Based on (18), the WEEI of each method is given by (19):

$$WEEI_i = \sum_{j=1}^4 w_j x_{ij}. \quad (19)$$

V. RESULTS AND DISCUSSION

Here, we report on the experiment outcomes, analyze the statistical data, and discuss the results based on the abovementioned four evaluation indexes: runtime, repeatability, matching precision, localization error, and a comprehensive index WEEI.

A. Comparison of feature point repeatability

Fig. 5 shows the feature extraction results of the six methods. We observe clusters with SURF and AKAZE, especially with SURF. The distribution of point features extracted by ORB+BEBLID and A-BEBLID is relatively even and distinguishable. The total amount of feature points extracted by SIFT and SURF is lower than the others. A-BEBLID captured more feature points in groups Scale, Rotation and Affine compared to ORB+BEBLID.



Fig. 5: Feature extraction results of the six methods.

As shown in Fig. 6, the repeatability of A-BEBLID and AKAZE is 64% and 65%, respectively. Following that is the ORB+BEBLID: 42%, and ORB: 38%; whereas SURF is 28% and SIFT is 27%, the lowest.

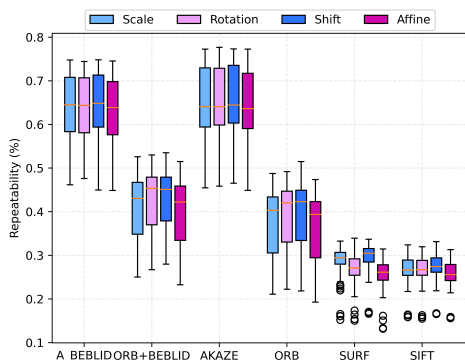


Fig. 6: Feature point repeatability of the six methods.

B. Comparison of matching precision

To improve matching precision, we applied GMS to remove the wrong matching pairs. The matching results on LSCSP and HPatches are shown in Fig. 7 and Fig. 8, respectively.

After applying GMS, we use the homography matrix to align the real image with the template image through the transformation matrix. We use the homography matrix obtained using the RANSAC algorithm by selecting the optimal four points of the two images. Then, we calculate the distance between the transformed coordinates of the real image and the coordinates of the template image, and use it as a threshold to calculate matching precision.

Fig. 9 and Fig. 10 show the average precision tested on the LBCSP and HPatches images, respectively. On the LBCSP dataset, the average precision of the proposed method is the highest when the threshold is ≤ 4 pixels. On the HPatches, the average precision of the proposed method is the highest when the threshold is 2 pixels. The ORB+BEFLID method is slightly higher than A-BEFLID in threshold range (2, 6).



Fig. 7: Matching results of the six methods on LBCSP.

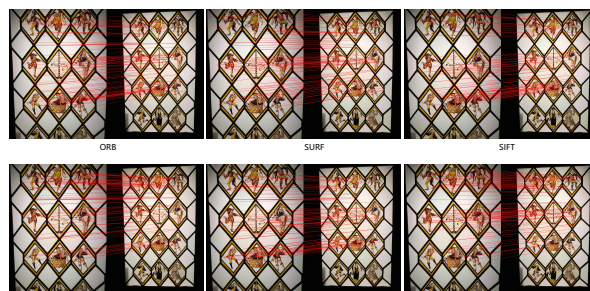


Fig. 8: Matching results of the six methods on HPatches.

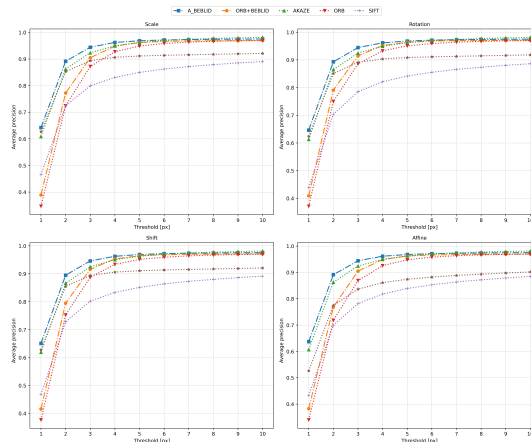


Fig. 9: Average precision of the six methods, tested on the LBCSP images.

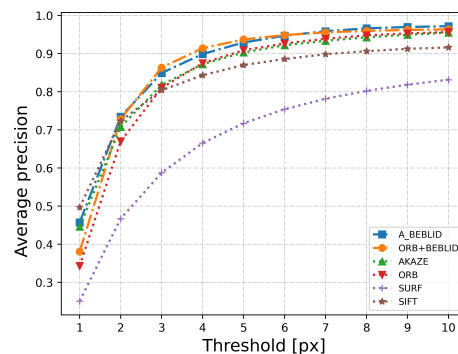


Fig. 10: Average precision of the six methods, tested on the HPatches images.

C. Comparison of runtime

The recorded runtime is the mean value of 392 runs in each group. The data is organized in Table I. In terms of average runtime, the "ORB+GMS" method obtained the fastest (highlighted in bold) of 0.262s, because of ORB's computational efficiency. Whereas "SURF+GMS" runtime is the slowest, 1.132s. The proposed method obtained a 0.348s (highlighted in Italian), ranks third.

D. Comparison of localization error

Fig. 11 shows the localization error of the six methods combined with GMS. The average localization error of SIFT

TABLE I: Runtime of the six methods combined with the GMS algorithm (s), the best result is highlighted in bold.

	SIFT +GMS	SURF +GMS	ORB +GMS	AKAZE +GMS	ORB+BEBLID +GMS	A-BEBLID +GMS
Scale	1.113	1.120	0.261	0.487	0.276	0.350
SD	0.610	0.578	0.031	0.254	0.028	0.179
Rotation	1.111	1.135	0.261	0.483	0.276	0.347
SD	0.607	0.587	0.032	0.252	0.027	0.177
Shift	1.110	1.165	0.262	0.489	0.276	0.352
SD	0.620	0.602	0.031	0.256	0.027	0.180
Affine	1.068	1.106	0.262	0.477	0.276	0.343
SD	0.566	0.569	0.032	0.247	0.029	0.173
Average	1.101	1.132	0.262	0.484	0.276	0.348

*Runtime is the mean value of 392 executions of each group.

is 1.07 pixels, the lowest of all six methods, followed by the A-BEBLID of 1.11 pixels. Whereas ORB shows the highest of 1.61 pixels, the ORB+BEBLID method shows the second highest of 1.52 pixels.

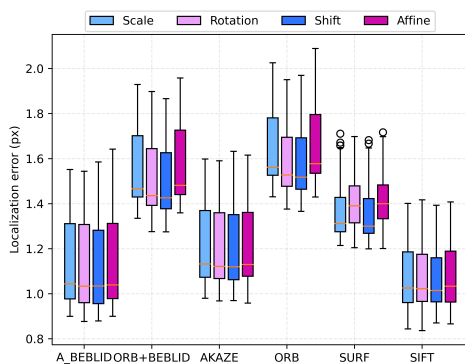


Fig. 11: Localization error of the six methods.

E. WEEI

The WEEI considers all the abovementioned four indexes: runtime, repeatability, matching precision, and localization error, and computes their weighted sum to evaluate their overall performance.

Table II and IV show the weight of each index calculated on LBCSP and HPatches, respectively. Table III and V show the overall evaluation result of the six methods tested on LBCSP and HPatches, respectively. A lower comprehensive score indicates better overall evaluation, as the negative index formula was used. The WEEI result verifies the effectiveness of the proposed method in scenarios where these four indexes are all taken into consideration.

TABLE II: Weight of each index on LBCSP.

	Runtime	Repeatability	Matching precision	Localization error
Weight	0.272	0.347	0.146	0.235

VI. CONCLUSION

During printing defects inspection, manufacturers face the challenges of image rotation and/or shifting caused by machine

TABLE III: WEEI of the six image registration methods on LBCSP.

	SIFT GMS	SURF +GMS	ORB +GMS	AKAZE +GMS	ORB+BEBLID +GMS	A-BEBLID +GMS
WEEI	0.818	0.907	0.672	0.540	0.641	0.488

TABLE IV: Weight of each index on HPatches.

	Runtime	Repeatability	Matching precision	Localization error
Weight	0.233	0.300	0.214	0.253

TABLE V: WEEI of the six image registration methods on HPatches.

	SIFT GMS	SURF +GMS	ORB +GMS	AKAZE +GMS	ORB+BEBLID +GMS	A-BEBLID +GMS
WEEI	0.837	0.922	0.697	0.671	0.647	0.625

vibration, which makes it more difficult to locate the desired features, resulting in miss-detection and false-detection. In this study, we collected 1568 images from the lithium-ion battery production lines and constructed a dataset for image registration research. After applying six different methods on the dataset and the public dataset HPatches, we evaluated the performance of these methods using feature extraction repeatability, matching precision, runtime, and localization error.

In terms of repeatability, the proposed method A-BEBLID and AKAZE are the highest, followed by the ORB and ORB+BEBLID. SURF and SIFT are the lowest.

In terms of matching precision, the average precision of the proposed method is the highest when the threshold is ≤ 4 pixels on the LBCSP dataset. While on HPatches, the proposed method is the highest when the threshold is 2 pixels. The ORB+BEBLID method is slightly higher than A-BEBLID in pixel range (2, 6].

In terms of runtime, ORB is the fastest in each test. This shows the superiority of ORB's computation efficiency. The proposed method is 0.086s slower than ORB, and 0.136s faster than AKAZE, while both SIFT and SURF took longer than 1.1s.

In terms of localization error, SIFT is the lowest at 1.07 pixels and the proposed method is 1.11 pixels, which ranks second.

Based on these results, we can see that the proposed method achieved the highest precision in our desired threshold range, and high rankings in the other three indexes. So we had a hunch that the hybrid method of AKAZE nonlinear-scale space structure and BEBLID binary descriptor is a suitable solution for our task. The WEEI overall evaluation score verified this hunch. By employing a nonlinear-scale space structure, the method improves the system's resistance to size change, rotation, and location shifting, which makes it suitable for the printing quality check of lithium-ion battery production lines. By adopting the BEBLID descriptor, the method increases the system's speed. It is worth noting that for the final step, we implemented the GMS algorithm to all six methods for matching pair selection to increase accuracy. The precision, runtime, and localization error are calculated after

feeding the matching results to GMS. With the rapid growth of computation power these days, accuracy will be a more important index in the industrial production field compared to runtime, and the evaluation balance will shift toward accuracy as the improvement of computation power naturally leads to an increase in speed.

REFERENCES

- [1] C. Liu, J. Xu, and F. Wang, "A Review of Keypoints' Detection and Feature Description in Image Registration," *Scientific Programming*, vol. 2021, p. 8509164, Dec. 2021. Publisher: Hindawi.
- [2] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep learning techniques for medical image segmentation: achievements and challenges," *Journal of digital imaging*, vol. 32, no. 4, pp. 582–596, 2019.
- [3] Y. Zhang, "Twenty-five years of image engineering in china," *Journal of Image and Graphics*, vol. 26, no. 10, p. 2326 – 2336, 2021. Cited by: 2.
- [4] A. A. Goshtasby, *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*. John Wiley & Sons, 2005.
- [5] W.-J. Zhang, F. Xu, X.-Y. Li, H. Xiao, S.-H. Peng, and H.-D. Nam, "Automatic printing plate defect detection based on a simplified homocentric square filter," in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 336–340, 2016.
- [6] H. P. Moravec, "Techniques towards automatic visual obstacle avoidance," 1977.
- [7] C. Harris, M. Stephens, et al., "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, pp. 10–5244, Citeseer, 1988.
- [8] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, 1999.
- [9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [10] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [11] D. G. Viswanathan, "Features from accelerated segment test (fast)," in *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*, pp. 6–8, 2009.
- [12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*, pp. 778–792, Springer, 2010.
- [13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, pp. 2564–2571, Ieee, 2011.
- [14] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE Features," in *Computer Vision – ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), (Berlin, Heidelberg), pp. 214–227, Springer Berlin Heidelberg, 2012.
- [15] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2011.
- [16] I. Suárez, G. Sfeir, J. M. Buenaposada, and L. Baumela, "BEBLID: Boosted efficient binary local image descriptor," *Pattern recognition letters*, vol. 133, pp. 366–372, 2020.
- [17] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [18] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5173–5182, 2017.
- [19] B. D. De Vos, F. F. Berendsen, M. A. Viergever, H. Sokootti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical image analysis*, vol. 52, pp. 128–143, 2019.
- [20] Z. Linna, C. Jianqiang, W. Yan, Z. Yue, and C. Yigang, "Registration algorithm of multi-repeat texture images based on double-match image registration," *Shuju Caiji Yu Chuli/Journal of Data Acquisition and Processing*, vol. 36, no. 2, p. 334 – 345, 2021.
- [21] Y. Chen, P. He, M. Gao, and E. Zhang, "Automatic feature region searching algorithm for image registration in printing defect inspection systems," *Applied Sciences (Switzerland)*, vol. 9, no. 22, 2019.
- [22] M. Zhou, G. Wang, J. Wang, C. Hui, and W. Yang, "Defect detection of printing images on cans based on ssim and chromatism," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2127–2131, 2017.
- [23] F. Hu. and H. Guo., "Printing defects inspection based on improved multi-class support vector machine," *Huadong Ligong Daxue Xuebao/Journal of East China University of Science and Technology*, vol. 43, no. 1, p. 143 – 148, 2017.
- [24] B. Liu, Y. Chen, J. Xie, and B. Chen, "Industrial Printing Image Defect Detection Using Multi-Edge Feature Fusion Algorithm," *Complexity*, vol. 2021, p. 2036466, Oct. 2021. Publisher: Hindawi.
- [25] Z. Dong., B. Liu., C. Hu., M. Gao., and P. Li., "Research on surface defect detection method of screen printing template based on machine vision," *Gaojishu Tongxin/Chinese High Technology Letters*, vol. 30, no. 12, p. 1309 – 1316, 2020.
- [26] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [27] G. Shakhnarovich, *Learning Task-Specific Similarity*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [28] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of computer vision*, vol. 37, no. 2, pp. 151–172, 2000.
- [29] Q. Li, H. Hu, L. Ma, Z. Wang, M. Arici, D. Li, D. Luo, J. Jia, W. Jiang, and H. Qi, "Evaluation of energy-saving retrofits for sunspace of rural residential buildings based on orthogonal experiment and entropy weight method," *Energy for Sustainable Development*, vol. 70, pp. 569–580, 10 2022.
- [30] C.-I. Gao, S.-c. Li, J. Wang, L.-p. Li, and P. Lin, "The Risk Assessment of Tunnels Based on Grey Correlation and Entropy Weight Method," *Geotechnical and Geological Engineering*, vol. 36, pp. 1621–1631, June 2018.



Na Wang (M'22) received her M.S. degree in Power Electronics and Power Drives from South China University of Technology. She is currently a Lecturer with the School of Automation, Guangdong Polytechnic Normal University. Her research interests include electronics, signal processing, and artificial intelligence. She is also a Ph.D student in the University of Newcastle, Australia.



Ying Chen received her M.S. degree in Control Science and Engineering from Guangdong Polytechnic Normal University in 2021. Her research interests include image recognition, defect monitoring and computer vision. She is currently the Director of MCU Training Centre of the Department of Intelligent Engineering in Guangdong Maoming Agriculture & Forestry Technical College.



Xianyong Zhang received his Ph.D. degree in Control Theory and Application from South China University of Technology in 2007. He is a Professor with the School of Automation, Guangdong Polytechnic Normal University. His research interests include new energy, power delivery, control theory and control engineering. He is the author of numerous publications in the form of books, journal articles, conference papers and invention patents.



(NSFC) and the Provincial Department of Education.

Xuhong Zhang received his Ph.D. degree in Electrical Engineering from Hunan University, China. He is an Associate Professor with the School of Automation, Guangdong Polytechnic Normal University. His research interests include smart grid, artificial intelligence and modern detection technology. He is the author of 2 books, 11 articles, 4 invention patents, and 10 utility model patents and software copyrights. He has participated in research projects funded by the National Natural Science Foundation of China



processing. To date, He has produced/co-authored over 200 refereed publications in the form of books, book chapters, journal articles and conference papers, and attracted more than 3 million dollars in research and industry funding. He is the Editor-in-Chief of the Journal of Systems and Information Technology, an Editor of Engineering Applications of Artificial Intelligence, and an Associate Editor of Engineering Reports.

Raymond Chiong (SM'14) graduated with a Ph.D. degree from the University of Melbourne, Australia. He is currently an Associate Professor with the School of Information and Physical Sciences, University of Newcastle, Australia. He has been actively pursuing research related to the use of automated intelligent computing methods, including machine learning and optimisation algorithms, with applications in energy load forecasting, energy-efficient production scheduling, computer vision and image processing.