Industrial Vision: Rectifying Millimeter-Level Edge Deviation in Industrial Internet of Things With Camera-Based Edge Device

Lei Xie, *Member, IEEE*, Zihao Chu, Yi Li, Tao Gu, *Member, IEEE*, Yanling Bu, *Member, IEEE*, Chuyu Wang, *Member, IEEE*, and Sanglu Lu, *Member, IEEE*

***Abstract*—Nowadays, to realize the intelligent manufacturing in Industrial Internet of Things (IIoT) scenarios, novel approaches in computer vision are in great demand to tackle the new challenges in IIoT environment. These approaches, which we call *Industrial Vision*, are expected to offer customized solutions for intelligent manufacturing in an accurate, time efficient and robust manner. In this paper, we propose a novel approach to industrial vision, called *Edge-Eye*, to rectify the edge deviation automatically for Irradiated Cross-linked Polyethylene Foam (IXPE) production with millimeter-level accuracy. We deploy a commercial camera with mobile edge node in front of the IXPE sheet to continuously detect and rectify the edge deviation. Particularly, to handle the complex production environment when extracting the edge of IXPE sheet, we deploy a pair of reference bars with high-contrast colors to efficiently differentiate the sheet edge from the background. Then, we propose a *Bi-direction Edge Tracking method* to perform the edge detection from both vertical and horizontal aspects. To realize the rectification using mobile edge nodes with limited computing resources, we reduce the cost of computation by extracting the *Minimized Region of Interest*, i.e., the edge area overlapped with the higher contrast reference bar on both sides. We further design a negative feedback control system with multi-stage feedback regulation mechanism, keeping the edge deviation within *millimeter-level*. The experimental results show that *Edge-Eye* achieves the average accuracy of 5 mm for the edge deviation rectification, with the average latency of 200 ms for edge deviation detection.**

***Index Terms*—Edge computing, edge deviation rectification, industrial vision, industial internet of things.**

Manuscript received 9 June 2022; revised 12 December 2022; accepted 8 February 2023. Date of publication 17 February 2023; date of current version 8 January 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3303900, in part by the National Natural Science Foundation of China under Grants 61832008 and 62272216, and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization. This work was also supported in part by Australian Research Council (ARC) Discovery Project under Grant DP190101888. Recommended for acceptance by D. Koutsonikolas. *(Corresponding authors: Lei Xie; Chuyu Wang)*

Lei Xie, Zihao Chu, Yi Li, Chuyu Wang, and Sanglu Lu are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: lxie@nju.edu.cn; zihaochu@smail.nju.edu.cn; yili@smail.nju.edu.cn; chuyu@nju.edu.cn; sanglu@nju.edu.cn).

Yanling Bu is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: yanling@smail.nju.edu.cn).

Tao Gu is with the School of Computing, Macquarie University, Macquarie Park, NSW 2109, Australia (e-mail: tao.gu@mq.edu.au).

Digital Object Identifier 10.1109/TMC.2023.3246176

I. Introduction

A. Motivation

As the proliferation of Industrial Internet of Things (IIoT) and Industrial 4.0, computer vision has been more and more widely used in the manufacturing scenarios, aiming to realize various functions including recognition, measurement, positioning and detection. However, in real complex scenarios of manufacturing, the traditional computer vision technology usually fails to achieve expected performance in accuracy, time efficiency and robustness. The reason is that, the traditional computer vision usually cannot provide customized operators to satisfy the special requirements in manufacturing, e.g., detection of appearance defect and measurement of complex forms. Besides, the manufacturing environments are usually full of various interferences and noises, e.g., the complex background, the scattering or occlusion of light. These issues greatly prohibit the traditional computer vision from efficiently performing the conventional functions in IIoT scenarios. Therefore, it is essential to explore novel approaches in computer vision to efficiently tackle the brand-new challenges appearing in IIoT scenarios. These kinds of approaches, which we call *Industrial Vision*, should be able to offer customized solutions to intelligent manufacturing in an accurate, time-efficient and robust manner.

Irradiated Cross-linked Polyethylene Foam (IXPE) is one of the fundamental industrial materials, widely used for the automotive trim, upholstery, and industrial packaging. During the production process, raw IXPE sheets will first go through a high temperature furnace via the conveyor belt to be heated, softened and foamed. When coming out from the furnace, they will be stretched and widened by a pair of spreader rolls, and rolled up into a roll. The roll will be finally trimmed from both sides to end the production. In this process, it is crucial to align both edges of IXPE sheets on the conveyor belt continuously. Since the conveyor belt moves at a speed of over 1 m/s, any misalignment could be quickly accumulated in the rolling up stage, causing more edges trimmed away and hence more materials wasted. However, it is difficult to stretch and widen IXPE sheets in a uniform manner due to the non-uniformity of thickness in raw IXPE sheets. Thus, the real edge deviation in the production line is ranged from 1 cm to 3 cm per second on average.

The existing edge deviation rectification relies on the human supervision, i.e., an operator monitors the IXPE sheet in real

time and adjusts the edge when necessary. This requires a high level of concentration from the operator, which is very labor-intensive. For a long time, the rectification precision of IXPE production cannot be guaranteed, resulting in a high level of material waste. As the Industrial Internet of Things (IIoT) is increasingly deployed in manufacturing, fully automatic edge deviation rectification can be made possible for IXPE production, greatly reducing the labor cost and effectively improving the long-term production efficiency.

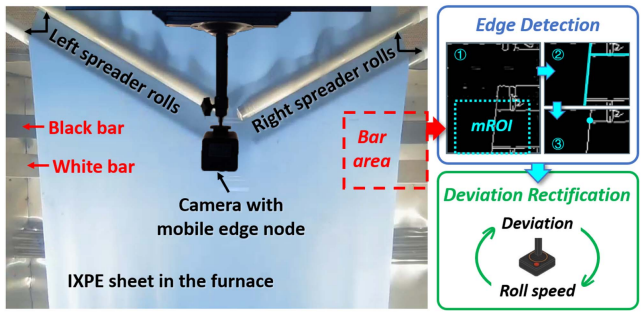


Fig. 1. Illustration of Edge-Eye in practical production line.

B. Limitations of Prior Art

Several solutions have been proposed to detect the edge position, including *laser ranging*, *millimeter wave*, and *camera*-based solutions. The laser ranging solution can accurately detect the depth difference between IXPE sheet and background, hence detect edges. However, as the laser unit detects the distance of one position at a time, it is inconvenient to deploy the laser ranging array for monitoring the whole furnace, or incurs delay if moving a single unit. Millimeter wave (mmWave) radar can achieve high accuracy for distance estimation, but not for angular estimation Thus, it is also unsuitable for detecting the material edge. Although the moving scanning of mmWave radar can improve the angular accuracy, the movement increases the detection delay [1] as well. Compared with laser and mmWave, computer vision can detect the contour edge, hence appropriate for edge detection. Especially, 3D cameras measure the depth for each pixel using Time-of-Flight (ToF) or structured light. However, 3D cameras can be costly for mass deployment, and the accuracy may degrade in the complex production environment.

Therefore, the design of edge detection and deviation rectification for IXPE production requires: 1) *Accurate:* the average error should be below 5 mm, 2) *Time-efficient:* the average response time should be below 200 ms, and 3) *Robust:* being able to perform the edge detection for different colors of IXPE sheets, 4) *Smooth:* the deviation rectification should be performed in an adaptive and steady manner.

C. Proposed Approach

In this paper, we propose a novel approach to industrial vision, i.e., *Edge-Eye*, a camera-enabled IoT edge device to automatically rectify the edge deviation for IXPE production with millimeter-level accuracy. Specifically, we use an ordinary camera running on the ARM64 platform as the Mobile Edge Node (MEN), as shown in Fig. 1. In order to detect the edge deviation, we first adaptively extract the *Minimized Region of Interest* (mROI) to reduce the computing cost. Further, we propose a *Super-Resolution-based Upsampling method* to construct a higher resolution image with edge points in finer granularity. Then, we use a *Bi-Direction Edge Tracking method* to achieve the highly accurate and reliable edge detection. To rectify the edge deviation, we propose a negative feedback control scheme with multi-stage feedback regulation to minimize the edge deviation to *millimeter-level*.

D. Challenges

There are three technical challenges in this paper. The first challenge is to *accurately* extract the edges of IXPE sheets from the image captured by the camera in a *robust* manner. Since the color of IXPE sheets may vary over time, it could be very close to the background color of furnace. The color similarity between the sheet and the background would cause the significant accuracy degradation for edge detection. To address this challenge, we deploy a pair of reference bars with high-contrast colors, i.e., white and black, under the conveyor belt, as shown in Fig. 1. In this way, the white bar and black bar will be used as auxiliary references to reduce the complex interference from background. Depending on the IXPE sheet color, *Edge-Eye* selects either the white bar or black bar to produce the best color contrast. Then, we propose a *Bi-Direction-based Edge Tracking method* to perform the edge detection vertically and horizontally. In the vertical direction, we detect the boundary between the IXPE sheet and selected bar; in the horizontal direction, we detect the leftmost or rightmost point for the uncovered part of selected bar. We fuse the two orthogonal results into a complimentary filter to figure out a more accurate edge position. In this way, we can guarantee the high *accuracy* and *robustness* by performing the bi-direction edge tracking with the high-contrast reference bars.

The second challenge is to monitor the edge deviation by only using the mobile edge device with limited computing resources in a *time-efficient* manner. The sensing delay comes from detecting and tracking the edge. To address this challenge, we propose to extract the *mROI* to sufficiently reduce the sensing delay, i.e., the edge area overlapped with the higher contrast reference bar on both sides. Moreover, the continuous edge detection usually consumes more computing resources, so we use the *cache-pool-based method* to reduce the repetitive computation. When the difference between the current frame and the cache frame is below a threshold, we directly reuse the previous results, otherwise, the current frame is set as the new cache frame, and the edge detection results are updated accordingly. In this way, the computing resources can be greatly reduced by shrinking the ROI in the space domain and reusing the edge detection results in the time domain.

The third challenge is to rectify the edge deviation in an *adaptive* and *steady* manner. The edge deviation rectification is done by changing the rolling speed of the left or right spreader rolls. However, it is quite difficult to figure out the uncertain

relationship between the edge deviation and the rolling speed. Thus, we propose a negative feedback control scheme, to formulate this relationship as a linear model on a small scale. When an edge deviation is detected, we use this model to calculate the speed change of spreader rolls and rectify this deviation until the edge goes back to the standard position. In this way, we can perform the edge deviation rectification in an *adaptive* manner. Moreover, considering that when large edge deviation happens, frequent adjustments in spreader rolls are essentially needed. Inappropriate adjustments will enlarge the jitters of edge position and decrease the rectification performance. To tackle this issue, we propose multi-stage feedback regulation mechanism to smooth the frequent adjustment of rolling spreaders. We leverage a sliding window to progressively approach the expected value from coarse granularity to fine granularity in a multi-stage manner. In this way, we can perform the edge deviation rectification in a *steady* manner.

E. Contributions

This paper makes the following contributions. First, we propose *Edge-Eye*, a millimeter-level edge deviation rectification system. To the best of our knowledge, *Edge-Eye* is the first system which uses computer vision and negative feedback control to rectify the edge deviation for IXPE production. Second, we propose an efficient edge deviation detection method, by incorporating a pair of high-contrast reference bars, mROI extraction and bi-direction edge tracking, to achieve the high accuracy, real-time response and robustness in the industrial production. Moreover, we design a negative feedback control system, and propose the multi-stage feedback regulation mechanism to rapidly and accurately make control decisions. Third, we implemented *Edge-Eye* and evaluated its performance in real IXPE production lines. Experimental results show that we achieve an average accuracy of 5 mm for edge deviation rectification, and an average latency of 200 ms for edge deviation detection. During the process of 20-month real deployment for 36 production lines, 66 manpower per day (90% of the overall manpower) has been saved, and the utilization rate of the IXPE material increases from 87% to 94%, indicating *Edge-Eye* can effectively reduce the cost and improve the benefit in intelligent manufacturing.

II. Related Work

A. Distance-Based Edge Detection

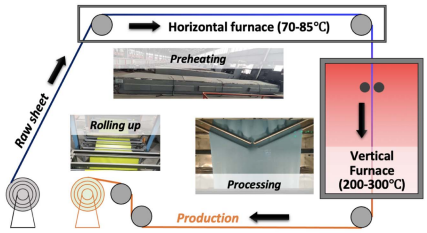
Since the target and other objects are in different planes, calculating the distance of all objects can accurately find the contour edge of target. Laser ranging accurately calculates the object distance by RTT method [2] but suffers from the single point measurement and accuracy decrease through glass. Millimeter wave radar measures the difference of frequency modulated continuous wave (FMCW) between TX and RX for distance calculation [3]. However, the angular resolution limitation makes it not suitable for the material edge detection. Moreover, the complex environment with serious multipath effect leads to low accuracy and high time latency [4] for the mmWave-based solutions. State-of-the-art technology, e.g., the 3D camera, can detect the edge position of an object by calculating the depth information for each pixel. Time-of-Flight (ToF) camera calculates the object depth by measuring the round trip time of an artificial light signal provided by a laser, but it suffers the same problem with laser ranging, e.g., accuracy decreases through glass and the edge blurs due to large scan intervals. Structured light camera uses the deformation principle to calculate the depth information, by analyzing the projection shift when the light hits the uneven surface of object [5]. This method suffers huge interference in strong light environment, the projected structured light can be easily submerged by strong light. Moreover, the accuracy decreases greatly when the object is 1 m away from the camera. Stereo camera uses the disparity to calculate object depth information by a pair of 2D cameras. However, it requires high computing resource and produces large error when environment is monotonous and lack of texture [6]. Therefore, in addition to the high hardware cost, different kinds of 3D cameras have their own limitation, making it unsuitable to use 3D cameras for the sheet edge detection in IXPE production.

B. 2D Camera-Based Edge Detection

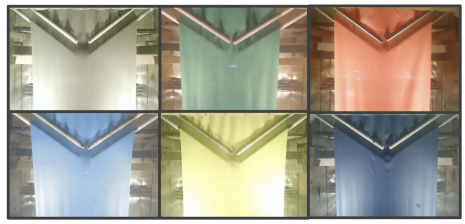
The edge is an inherent property of object, which usually has a sharp change of color around in a digital image. There have been a wide range of approaches to extract edges in images captured by 2D cameras. Edge detection aims to find the discontinuities of digital images, by finding the image points with great gradient [7].

Traditional Edge Detection Methods: Basic edge detection filters such as Sobel, Prewitt and Roberts calculate edge points by directly evaluating the pixel value difference of adjacent points in grayscale images [8], [9], [10], but such methods have fatal limitations of noise pollution and rough edges. Advanced edge detection operators, e.g., Canny operator and Marr-Hildreth operator, achieve high performance on edge points calculation. The former is a multi-stage algorithm [11], which uses image smoothing, intensity gradients calculation, double threshold and hysteresis to find the optimal edges in image. The latter uses second derivative and zero crossing to find edge points, but it is noise sensitive due to the second derivative process [7]. Both operators extract all possible edge points by relying on the original contrast of edge itself, however, they suffer from low material sheet edge extraction performance when similar color appears between the sheet and background.

CNN-Based Edge Detection Methods: With the popularity of Convolution Neural Networks (CNN), edge detection has been revisited and new solutions are proposed with neural networks [12], [13], [14]. Deepedge [12] designs a multi-scale deep network to achieve contour detection by using the object-related features as high-level cues. Maire et al. [14] use the generic deep sparse code to recognize specific targets, thus achieving target edge detection. Holistically-nested edge detection (HED) [13] uses the image-to-image training method to construct the representation network of original images and predicts edges. Casenet [15] uses ResNet and skip-layer architecture to realize category-aware semantic edge detection. Poma et al. [16]



(a) IXPE material production line: from raw material sheet to production sheet



(b) Complex background: IXPE sheets with various colors

(c) Manual rectification by human supervision

(d) Rectified IXPE material

(e) Unrectified IXPE material

Fig. 2. IXPE production process and unsatisfied performance of traditional edge operators.

propose thin edge-maps extraction by adding an upsample block in Dense Extreme Inception Network. Although CNN-based solutions can be more accurate than traditional methods, they consume more computing resources and additional expenditure on neural network training and storage. In addition, when the color of the material sheet is similar to the background, their edge extraction performance is still not high.

III. Preliminary

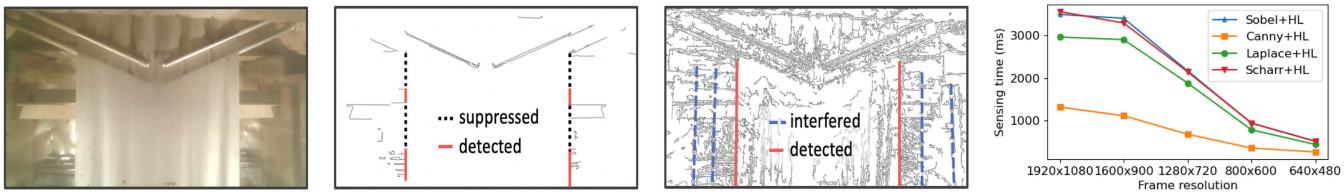
The foaming technology has been used in modern manufacturing to heat raw materials and form the required shape or size for end products. As shown in Fig. 2(a), the raw material sheet is first sent into the horizontal furnace (with a temperature of 70-85°C) for preheating, then goes into in the vertical furnace (with a temperature of 200-300°C) for softening, widening and stretching into the required size. Finally, the sheet is rolled up and packed as the finished sheet ready for shipment. The color of sheet can be various (in Fig. 2(b)), leading to a more complicated background for edge detection. Moreover, due to the uneven thickness of the raw material sheet, the width of finished sheet may be different from the required size, i.e., edge deviation. Without the proper rectification, such deviation may accumulate rapidly, resulting that more edges have to be trimmed away and hence more materials will be wasted. Traditional solutions rely on the human supervision, as shown in Fig. 2(c). An operator monitors sheet edges and adjusts the rotation speed of spreader rolls to align the edge. This solution typically has low accuracy in edge detection (5 cm) and large response time (5 s). Besides, it is difficult to train operators and assure quality for long-term production. If the IXPE material is not rectified by the spreader rolls in time, the IXPE material could fold quickly during the rolling process and turn into waste product. This could further lead to unpredictable losses for the IXPE production. Figs. 2(d) and (e) show the examples of the rectified IXPE material and unrectified IXPE material during the production process, respectively. Therefore, the goal of automatic rectification is to detect edge position accurately and timely in a complicated background.

As one of the best operators for edge detection, Canny operator extracts optimal edge points by double threshold (H_{high}, H_{low}) method [11]. Specifically, after calculating the gradient value of each edge pixel, the value higher than H_{high} is marked as a strong edge pixel, and the value lower than H_{low} gets suppressed. If the value is between H_{low} and H_{high}, it is marked as a weak edge pixel and turns to a strong edge pixel if connected with an original strong edge pixel, otherwise the weak edge pixel gets suppressed. Finally, all the strong edge pixels get output as edge points. However, Canny operator with fine-tuned parameters does not achieve the satisfactory performance on sheet edge extraction in following aspects.

1) *Accuracy and Robustness:* It is unable to achieve the accurate and robust edge detection performance just by adjusting parameters in Canny operator, not to say other less accurate edge detection methods. Specifically, Fig. 3(a) shows the original image of sheet edge position with a similar color in the background. With a high value for double threshold, Canny operator suppresses target edge points (the black dashed lines), resulting in the inaccurate edge position calculation in Fig. 3(b). While with a low value for double threshold, the result in Fig. 3(c) gives more interfered edges, e.g., interfered edges around target edge lines and blue dashed lines have similar features, leading to poor accuracy and failure.

2) *Time-Efficiency:* The edge deviation rectification for material sheet production requires real-time response, i.e., less than 200 ms. Traditional edge detection methods fail to meet this requirement as shown in Fig. 3(d). For Canny operator with Houghlines (HL) detection, when the frame size reduces from 1920×1080 to 640×480, the time delay reduces from 1,400 ms to 360 ms. Other edge operators even require larger processing time. Moreover, the low resolution image is not conducive to manual re-examination.

The above observations motivate several ideas in designing an accurate and time-efficient edge detection, which are summarized as follows: 1) To enhance accuracy, we should fuse



(a) Captured image (b) High value for double threshold in Canny operator (c) Low value for double threshold in Canny operator (d) Sensing time of operators in MEN

Fig. 3. Performance of various operators. (a)∼(c): accuracy and robustness, (d): time-efficiency.

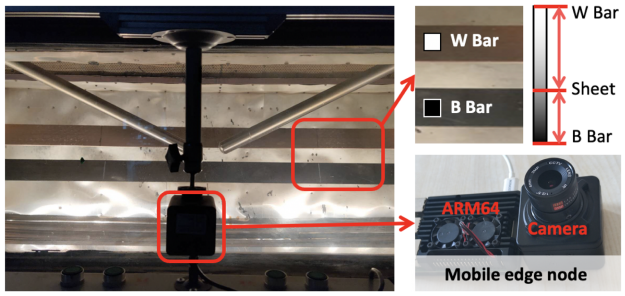


Fig. 4. System deployment.

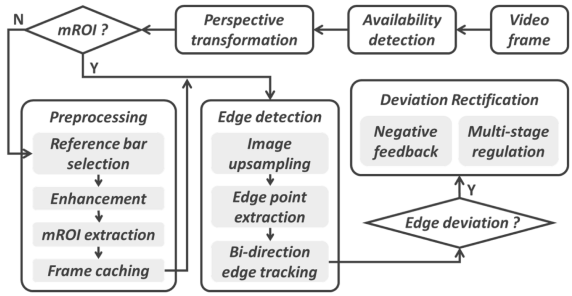


Fig. 5. System overview.

the results from multiple approaches; 2) To improve robustness, we should enhance the contrast between the sheet and background; 3) To achieve time-efficiency, we should focus on the specific region of interest instead of the full frame.

IV. System Design

System Deployment: Fig. 4 illustrates the deployment of *Edge-Eye*. We deploy a camera in front of the vertical heating furnace at a distance of 1.2 m from the target material sheet. Without loss of generality, the camera has a wide-angle lens of 120° with video quality of 1080p/30fps, thus each pixel of the image frame represents a width of 1.87 mm on the sheet plane. The images obtained from the camera contain much noise due to the color similarity between the sheet and the background. To minimize background noise, we use a pair of *reference bars* to generate high edge contrast between the material sheet and the background, while narrowing the observation range to a relatively clean and controllable area. Since the edge calculation relies on the difference of colors in the image, which can be transformed to the distance of gray-scale image. Here, we use the max/min gray values (0 and 255), which are exactly the white and black colors. As a result, either of the two bars (W Bar and B Bar) can always have a color difference no less than 255/2. To reduce the interference of light reflection in the recognition area, we set a *shading plate* in front of the camera to maintain a stable ambient light. We combine the camera with the ARM64 computing platform as our *mobile edge node* to detect sheet edge with fast response and low transmission delay.

Software Framework: Fig. 5 shows the modules of *Edge-Eye*. Specifically, 1) *Availability Detection* module checks whether any sheet is in production. This is done by comparing the difference between the current frame and the empty frame. 2) *Perspective Transformation* module corrects image distortion of captured frame and obtains straight edges. If the system is first running, i.e., no mROI is extracted, we run the *preprocessing* module to finish the initialization. 3) *Preprocessing* module selects the reference bar and extracts the mROI. It selects the highest contrast reference bar from W Bar and B Bar according to sheet color, and extracts the area near the edge of material sheet as mROI to reduce the computational overhead. Besides, it sets cached mROI to further speed up the detection of sheet edge. 4) *Edge Detection* module generates the super-resolution image for mROI, and provides the accurate sheet edge position by the bi-direction edge tracking method with abnormal detection. First, it uses the Fast Efficient Sub-Pixel Convolutional Neural Network (Fast-ESPCN) to build the super-resolution image from the original low-resolution image on this specific ROI area, thereby generating more edge points and finer grit description for material sheet. Second, it calculates the edge points and divides those points into background edge points and material sheet related edge points. Third, it tracks the sheet edge in vertical direction and uncovered part of reference bar in horizontal direction separately, and fuses two recognition results through the complimentary filter. Then, it detects and repairs abnormal results to realize high-precision edge position recognition. 5) *Deviation Rectification* module rectifies the edge deviation and adjusts sheet edge to the standard position. It uses the negative feedback control method to make a rapid and appropriate decision. We apply a linear model to depict the relationship between the sheet position and the speed of spreader rolls, thus we can keep sheet edge deviation within the standard range by adjusting the roll speed efficiently. We then use the multi-stage feedback regulation mechanism to dynamically adjust the parameters of the linear model, achieving a smaller deviation jitter range and shorter rectification time.

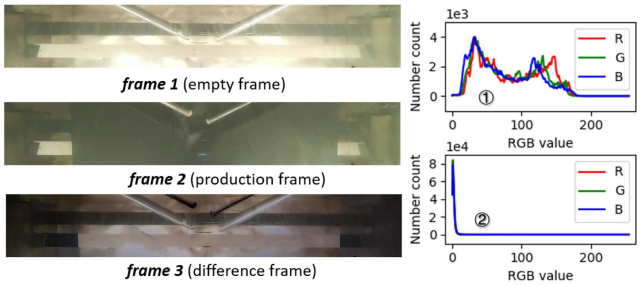


Fig. 6. Material availability detection around reference bars. Left: Frames in detection area (*frame 3 = frame 1 - frame 2*), Right: RGB distribution of: ① frame 3, ② difference frame between empty frames.

A. Availability Detection

The system should be triggered only when the IXPE production is in progress. To perform the material availability detection, we use the frame difference method to verify whether there exists material sheet on the conveyor. Specifically, we set up a detection area in the center of reference bars. As shown in Fig. 6(a), we record an empty background frame as the empty frame, and calculate the difference between the current frame and the empty frame. We thus perform analysis on the RGB distribution of the difference frame. As shown in Fig. 6(b), we can find that, in regard to RGB distribution, the difference between the empty frames is small, whereas the difference between the empty frame and the production frame is quite large. Therefore, we use the entropy of difference frame to determine whether there exists material sheet in production. Specifically, we calculate the entropy of R, G, and B channels in the difference frame F, respectively. Taking channel R as an example, we use h_i to denote the total number of pixels with value i in channel R, and use n to denote the number of different values of i in this channel. We use p_i to represent the ratio of pixels with value i to all pixels in channel R, thus $p_i = h_i/(\sum_{j=0}^{n-1} h_j)$. Then, the entropy value $H(F)$, which describes the information of channel R in the difference frame F, can be calculated as follows:

$$H(F) = -\sum_{i=0}^{n-1} p_i \log p_i. \tag{1}$$

To reduce the complexity in the calculation, we convert the RGB difference frame into a gray-scale image and reduce the value range from (0,255) to (0,16) by dividing point value by 16. Then we calculate the entropy of this simplified gray-scale image. To enhance environmental adaptability, we set two thresholds, H_{update} and $H_{material}$. For each periodic interval, e.g., 1 minute, when the $H(G)$ of current frame is less than H_{update}, we update the empty frame with the current frame. When $H(G)$ of the current frame is greater than $H_{material}$, it can be determined that the material sheet is in production and the edge detection should be started.

B. Perspective Transformation

A camera usually suffers the lens distortion when capturing images in real complex environment. Since the camera is deployed close to the sheet, this distortion will make the edge in captured frame bend seriously, which brings great interference to detection accuracy. Fortunately, the distortion is an inherent property of camera and all frames captured by one camera can be calibrated with the same calibration parameters. Therefore, before the camera is deployed, we use the chessboard-based calibration method to calculate parameters and correct distortion [17]. Specifically, the transformation relationship between distortion coordinate (x_c, y_c) and correction coordinate (x_p, y_p) is shown in (2), where $r^2 = x_p^2 + y_p^2$.

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} 2p_1 x_p y_p + p_2(r^2 + 2x_p^2) \\ 2p_2 x_p y_p + p_1(r^2 + 2x_p^2) \end{bmatrix}. \tag{2}$$

The parameters k_1, k_2, k_3 and p_1, p_2 represent the distortion factors, which can be calculated by the following steps: 1) use the camera to take photos of the chessboard from various angles, 2) search the corners of black and white boxes on those chessboard photos, 3) calculate the correspondence between the corners in the image and the real world, and generate spatial points in world coordinates, 4) calculate the corresponding camera parameters (k_1, k_2, k_3 and p_1, p_2) for camera calibration. Fig. 7 shows an example of perspective transformation on the captured distorted frame.

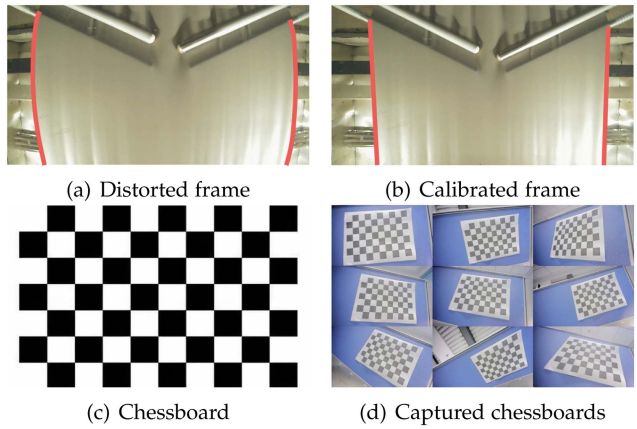


Fig. 7. Perspective transformation for captured frame.

C. Preprocessing

1) Reference Bar Selection: To obtain the highest image gradient at the edge of material sheet, we select the bar with higher contrast from W Bar or B Bar. Specifically, we use the following steps to calculate the score of each reference bar and select the one with higher score. First, we convert the RGB frame into a gray-scale image using (3), where R, G, B are the values of Red, Green, and Blue for each pixel in this RGB frame.

$$G_r = 0.2989 \times R + 0.587 \times G + 0.114 \times B. \tag{3}$$

Second, we make convolution using Sobel Filter in (4) and derive the gradient along x-axis (horizontal direction).

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}. \tag{4}$$

Third, we use a sliding window of size $m \times n$ to scan the gradient image with the step of η pixels, and calculate the maximum value of each row in the sliding window as the gradient list. Note that, the boundary between the material sheet and the reference bar generally corresponds to the area with high gradient values, hence we can determine the boundary area by comparing gradient lists of different sliding windows. Since the sheet has two edges, i.e., left edge and right edge, we search for the left and right boundary areas for each bar, respectively. Denote the average and standard deviation of each gradient list as μ and σ. We first select the sliding window with largest μ, and remove all overlapping windows. Then we select the second window with largest μ among the remaining. Actually, the two windows contain the left edge and right edge, separately. Assume the average and standard deviation of the two windows for one bar are μ_1, σ_1 and μ_2, σ_2, respectively. Thus, the score of one bar is calculated as

$$s = \mu_1 + \mu_2 - |\mu_1 - \mu_2| - \ln(\sigma_1 + 1)(\sigma_2 + 1). \tag{5}$$

The bar with higher score will be selected for reference.

2) Contrast Enhancement: To reduce the random interference from ambient noise and improve the quality of edge extraction, we perform the image enhancement for the boundary area of selected bar. Specifically, we use a sliding window of size $m \times n$ to scan the gradient image. When we find the parameter μ of the sliding window is larger than the preset threshold, for these areas, we use Laplacian Filter in (6) to enhance edge contrast.

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{6}$$

For other areas, we use Gaussian Kernel Filter in (7) to smooth random noise and suppress edge contrast.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{7}$$

3) mROI Extraction: To achieve time efficiency in edge detection on MEN with limited computing resources, we extract the mROI from the selected reference bar area. By removing the edge irrelevant areas, we are able to reduce recognition area by over 100 times compared with the full frame, as shown in Fig. 8. According to the continuity of sheet movement, we observe that the edge position changes slightly between adjacent frames. Thus we use the previous edge position to determine current mROI. After obtaining the left and right edge position of the previous frame, i.e., $x_l(t-1)$ and $x_r(t-1)$, we can extract the corresponding mROI areas of both edges for current frame as follows. Specifically, since the edge of reference bar helps determine the sheet edge, we set the height of mROI to a bit larger than the height of reference bar, so as to ensure that the extracted area contains the edge of reference bar. Meanwhile, the width of mROI depends on two factors: the edge position in the previous frame, i.e., $x_l(t-1)$ for left mROI and $x_r(t-1)$ for right mROI, and the searching range of $2x_\sigma$. Moreover, we extend the edge of mROI at the bar side with additional length of τ_x, to guarantee that enough reference bar and material edges are inside the mROI. In this way, we can improve time efficiency in the space domain.

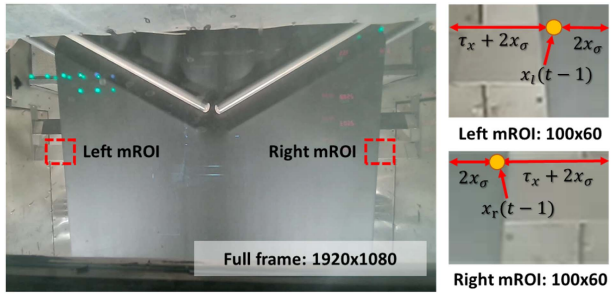


Fig. 8. mROI extraction.

4) Frame Caching: During the actual production process, due to the continuity of sheet movement, the recognition result of adjacent frames in the mROI usually tends to be consistent. Much time could be wasted in recognizing these frames repeatedly. Therefore, to achieve time efficiency, we propose cache-pool-based method to avoid the duplicated edge detection and further speed up the recognition process. Specifically, this pool records a fixed number of frames in the mROI. We use the Least Recently Used (LRU) based method [18] to update the cache pool. We define the cache hit as the entropy value of the difference frame between the current frame and the cached frame. If the entropy value is less than the *skip threshold*, the cache hit is successful. We then use the recognition result for the cached frame to skip the complicated edge detection. Otherwise, we start the normal edge detection process, and use the corresponding recognition result of the current frame to update the least recently used one in the cache pool. In this way, we can further improve the time-efficiency in the time domain.

D. Edge Detection

1) Image Upsampling: Compared with low-resolution images, high-resolution images provide more details in texture and help to improve the accuracy in edge detection. For example, the resolution in the 4 K image is four times higher than that of 1080p image. However, it is unaffordable for the limited computing platform MEN to capture and recognize the original 4 K resolution frames, along with providing the Real-Time Messaging Protocol (RTMP) media service. Besides, the cost of a camera that supports 4 K resolution is over four times larger than that of an ordinary 1080p camera. Therefore, according to the frame in the mROI captured by an ordinary 1080p camera, we leverage the *Fast-ESPCN upsampling* technology to generate a super resolution image. Compared with recognition on original 4 K frame, our method can achieve the equivalent edge detection accuracy but require much less recognition time and fewer computing resources.

Fast-ESPCN Structure Design. Inspired from the ESPCN model [19], we adjust some layers to achieve the fast speed

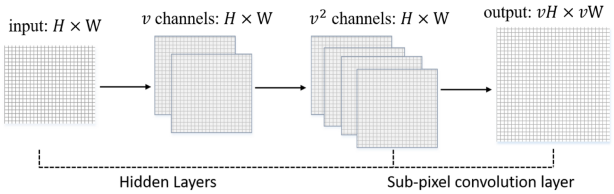


Fig. 9. Fast-ESPCN method takes the mROI image (low resolution) as input, and outputs the SR (super resolution) image.

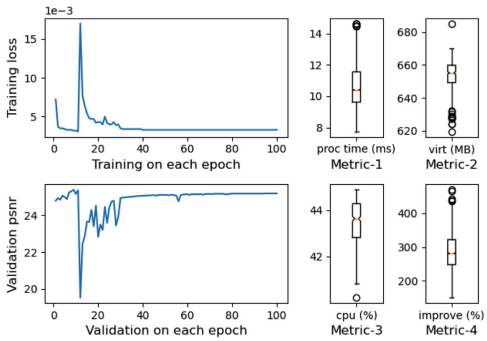


Fig. 10. Training/ Validation performance of Fast-ESPCN.

of training and inference, as shown in Fig. 9. For the frame in mROI with size $H \times W$, we use the growth type of hidden layers to get a v^2 (upsample factor) channels feature map with the same size. Then, a sub-pixel convolution layer is used to reshape the $H \times W \times v^2$ feature map to high resolution frame with size $vH \times vW \times 1$. Empirically, we set $v = 2$ to upsample the 1080p resolution image to 4 K.

Training and Validation. In the training stage, we first collect the original distorted 4 K frames, and use the perspective transformation to get calibrated 4 K frames as ground-truth. Then, we resize the calibrated frames to 1080p resolution as input. An NVIDIA GPU is used to complete the model training in offline mode, then MEN runs this model to upsample the mROI from 1080p low-resolution image to 4 K high-resolution image. Fig. 10 shows the benchmark performance of Fast-ESPCN module. With the training loss of 0.042 and validation Peak Signal-to-Noise Ratio (PSNR) of 26.3, our model achieves fast training convergence speed and high quality in image reconstruction. The results show that the Fast-ESPCN method achieves average processing time of 10.3 ms with a standard deviation of 4 ms, and the single CPU core utilization rate is 50%. Compared with the original downward type of hidden layers (105 ms average processing time and 27.1 PSNR), i.e., the hidden layer with the same hyperparameters set as the original ESPCN, with the same upscale factor ($v = 2$), our method achieves similar PSNR error but runs 10 times faster than the former.

2) Edge Point Extraction: With the high contrast between the material sheet and reference bar, the edge detection operators, e.g., the Canny operator with fine-tuned parameters can achieve good performance on the edge point extraction, as shown in Fig. 11. However, it is actually difficult to fine-tune parameters for each sheet color, and there are no static parameters that apply to all sheet colors. Based on this understanding, we aim

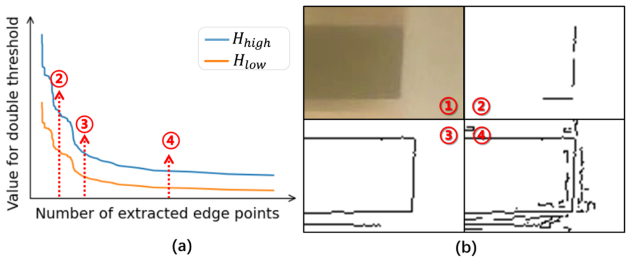


Fig. 11. Pre-study for Canny operator: (a) edge extraction performance for double threshold, (b) ① original mROI frame, ② large thresholds obtain fewer edge points, ③ appropriate thresholds obtain satisfying edge points, ④ small thresholds obtain noisy edge points.

to adaptively adjust parameters for Canny operator to accurately extract the two vertical lines for edge detection in the recognition area. Therefore, as mentioned in Section III, since the double thresholds of Canny operator, i.e., H_{low} and H_{high}, is very crucial to the edge detection, it is essential to obtain optimized values for the two parameters to improve the performance. We observe that, as shown in Fig. 8, for both the left and right edges of the material sheet in the mROI, they form a line with length no less than the height of mROI, respectively. We can use this property to evaluate whether the extracted edge points are satisfied for edge detection. Specifically, we set large values for (H_{high}, H_{low}) initially, it enables us to extract left and right vertical lines for the edge detection, where both lines have a small number of edge points to start with. Then, we iteratively update the values of (H_{high}, H_{low}) by step $(-H_u, -2H_u)$, and evaluate whether the extracted edge points form two major lines which are long enough. The iteration keeps running until the number of extracted edge points is greater than a certain threshold τ.

After fine-tuning the parameters of Canny operator, we extract all candidate edge points, including the background-related edge points and the sheet-related edge points. Thus, it is essential to separate the sheet-related edge points from the background-related edge points. We observe that, during the production process, only the sheet-related edge points move side to side, whereas the background-related edge points keep static. Therefore, we can divide all candidate edge points into static points, i.e., background-related edge points, and moving points, i.e., sheet-related edge points, from a time-domain perspective. Based on this understanding, we can calculate the intersection of edge points in multiple frames to extract the static points S_s, which keep static across multiple frames. Specifically, suppose the candidate edge points calculated by the Canny operator is S_i for each frame i, then, for the previous k frames, we define the common background edge points S_u as follows: $S_u = S_{i-k} \cap S_{i-k+1} \ldots \cap S_{i-1}$. After that, the static points S_s can be calculated as follows: $S_s = S_u \cap S_i$. Then, the moving points for each frame i can be calculated by subtracting the static points S_s from the candidate edge points, i.e., $S_m = S_i - S_s$. This separation results are shown in Fig. 12. Herein, the moving points S_m contain both the edge points of sheet as well as the occasionally blocked edge points of reference bar.

3) Bi-Direction Edge Tracking: After obtaining the edge points of sheet, we can perform the edge tracking for each

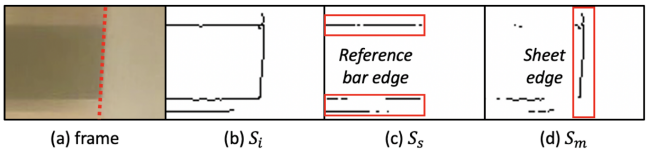


Fig. 12. Separate static and moving edge points for current image frame: (a) original mROI frame, (b) extracted edge points, (c) separated static edge points, (d) separated moving edge points.

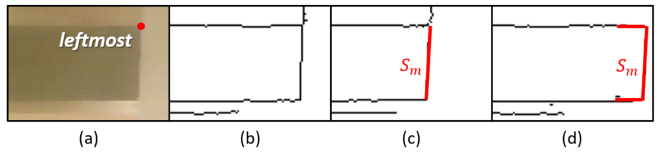


Fig. 13. Horizontal edge tracking for selected reference bar: (a) original frame of left mROI, (b) edge points from original frame, (c) sheet moves left, (d) sheet moves right.

Algorithm 1: Bi-Direction Edge Tracking.

Input: S_i: edge points set for current frame i; S_u, S_s, S_m: common background edge points, separated static points and moving points for S_i; $x(t-1)$: previous position of left/right sheet edge; τ_v: threshold for vertical line length;

Output: $x(t)$: current left/right sheet edge position;

Vertical Line Tracking:
Project S_m on y-axis as Y_m, count the number N of Y_m;
if $N < \tau_v$ **then**
  for $p = (x, y)$ *in* S_m **do**
    if $p' = (x \pm 1, y-1)$ *in* S_s **then** $S_m = S_m \cup p'$;
  end
end
Find an optimal vertical line in S_m, calculate $x_v(t)$;
Horizontal Line Tracking:
if *size(* $S_u \cap S_i$*)* <*size*(S_u) **then** Find optimal horizontal lines in S_s, calculate $x_h(t)$;
else Find optimal horizontal lines in S_m, calculate $x_h(t)$;
Bi-direction Fusion:
$x(t) = fusion(x_v(t), x_h(t), x(t-1))$;
return $x(t)$.

frame by searching vertical lines according to the extracted edge points. However, the edge tracking in a single direction, e.g., searching for edges in the vertical direction, is susceptible to the interference from ambient noise and may lead to the inaccurate recognition result. To improve the accuracy and robustness for edge tracking, we propose a *Bi-direction Edge Tracking* method. As shown in Algorithm 1, the algorithm is composed of three parts, i.e., *vertical edge tracking for material sheet*, *horizontal edge tracking for reference bar* and *bi-direction fusion*. After extracting edge points, we divide each set of edge points into left-edge-points and right-edge-points according to coordinates in the x-axis. Without loss of generality, we take one side of edge points as an example to show the detailed algorithm design of *Bi-direction Edge Tracking*.

Vertical Edge Tracking: In the vertical edge tracking, we aim to find a vertical line to accurately estimate the edge position of sheet, according to the extracted moving points S_m. However, when the sheet edge moves from side to side during the production process, the edge points of sheet can coincide with the static background edge points, resulting in part of missing sheet edge points among the extracted moving points S_m when performing $S_m = S_i - S_s$. Therefore, to identify the vertical line corresponding to the sheet edge, we need to verify and search for missing sheet edge points. Specifically, to verify if there exist missing sheet edge points in the vertical direction, we first project the moving points on y-axis and count the cardinality of unique projected points. If the cardinality is less than a threshold τ_v, we then use the static points vertically adjacent to the moving points to recover the originally missing sheet edge points. After that, we check all vertical lines with length greater than threshold τ_v from the updated moving points S_m, and identify an optimal vertical line corresponding to $x_v(t)$ to denote the sheet edge.

Horizontal Edge Tracking: In the horizontal edge tracking, we aim to find a leftmost or rightmost point of selected reference bar to accurately estimate the edge position of sheet. Herein, the leftmost or rightmost point corresponds to the boundary between the reference bar and the sheet, respectively. In principle, the leftmost or rightmost point can be identified from the static edge points S_s. However, when the sheet edge moves from side to side as shown in Fig. 13, the static edge points of selected reference bar can be occasionally blocked by the sheet, causing the leftmost or rightmost point to be possibly categorized to moving points. Therefore, we need to further identify the leftmost or rightmost point from either the static points or moving points. Specifically, when the sheet edge moves towards the center position, more points of selected reference bar can be extracted. In this situation, we search the horizontal line from S_m and use the leftmost/rightmost point of the line as the recognition result. When the material sheet moves away from the center position, fewer points of reference bar get extracted. In this situation, for the current frame i, we search the horizontal line from S_s, and use the leftmost/rightmost point of the line as the recognition result $x_h(t)$.

Bi-Direction Fusion: According to the recognition results from *Vertical Edge Tracking* and *Horizontal Edge Tracking*, it is difficult to determine which one is more accurate and reliable when the two relatively independent results are different. Considering the continuity of sheet movement, we give different weights on current horizontal/vertical edge tracking results. Specifically, we set the weight according to the difference between the previous fused result and the current horizontal/vertical edge tracking results. In the complimentary filter, the one with closer distance to the previous fused result will have higher weight, as shown in (8).

$$x(t) = \frac{|x_h(t) - x(t-1)| \times x_v(t) + |x_v(t) - x(t-1)| \times x_h(t)}{|x_v(t) - x(t-1)| + |x_h(t) - x(t-1)|}. \tag{8}$$

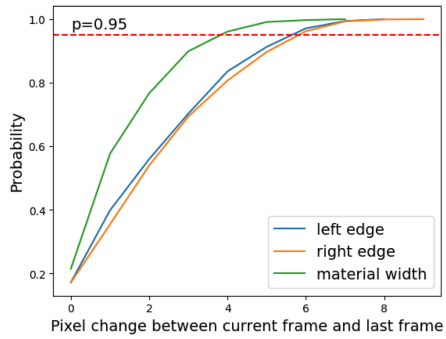


Fig. 14. CDF of edge position change and width change.

Herein, $x(t)$ and $x(t-1)$ denote the fused results at current time t and previous time $t-1$, respectively. $x_v(t)$ and $x_h(t)$ denote the recognition results from vertical direction and horizontal direction at time t. The weight of current horizontal edge tracking result $x_h(t)$ is $|x_v(t) - x(t-1)|$, which is the difference between current vertical edge tracking result and previous fused result, and the weight of current vertical edge tracking result $x_v(t)$ can be calculated similarly. Based on the complimentary filter, we can derive the accurate result with stable and smooth property in the time domain.

4) Abnormal Results Detection: In actual production process, if the detected edge position deviates too much from the actual position, the incorrect manipulation in the following deviation rectification could cause the material sheet to shift dramatically. The material sheet could even shift out of the production position, leading to the forced interruption of automated production. We find that, since the material is maintained at a relatively fixed width and the edge position changes slowly during normal production process, the change of material sheet's edge position should be small between current frame and last frame. To ensure the stability of automated production, we use the continuity of sheet width change in time-domain to detect abnormal results in edge position recognition. By calculating the interquartile range (IQR), we determine whether current recognition result is abnormal or not. The result is accepted if the width change is smaller than a certain threshold, e.g., the pixel change corresponding to a specified probability, say 95%, as shown in Fig. 14. Otherwise, we use the most likely value calculated from recent data to replace this result and ensure the stability of automated production.

Specifically, to calculate the most likely value according to recent data, we define the material width $w(t) = x_r(t) - x_l(t)$, and the width change can be represented as $\Delta w(t) = w(t) - w(t-1)$. First, we calculate the calibrated material width $\widetilde{w}(t)$ according to the linear regression in the previous n samples. In this way, we obtain the width which accords with the variation trend of the width in time-domain. Then, we use this calibrated width $\widetilde{w}(t)$ to calculate the calibrated left edge position $\widetilde{x_l}(t)$ and right edge position $\widetilde{x_r}(t)$, as shown in (9).

$$\begin{aligned} &\underset{\widetilde{x_r}(t),\widetilde{x_l}(t)}{\arg\min} \left(\widetilde{x_l}(t) - x_l(t)\right)^2 + \left(\widetilde{x_r}(t) - x_r(t)\right)^2 \\ &\text{subject to } \widetilde{x_r}(t) - \widetilde{x_l}(t) = \widetilde{w}(t). \end{aligned} \tag{9}$$

E. Deviation Rectification

The edge deviation appears when the detected edge position is not at the standard position. Once the edge deviation exceeds a certain threshold, it is essential to efficiently rectify the sheet edge deviation. Traditionally, the edge deviation rectification is conducted manually. After the material sheet expands to the product size, the workers will adjust the rolling speed of stretching devices, i.e., the left and right spreader rolls, to rectify the sheet edge deviation when necessary and keep the sheet edge at the standard position. For example, when the edge position deviates to the left side of standard position, the workers try to gradually increase the rolling speed of left spreader rolls or reduce the rolling speed of right spreader rolls, until the sheet edge returns to the standard position. The adjustment of rolling speed is purely determined by human experience. However, to perform the edge deviation rectification automatically, it is rather difficult to figure out the uncertain relationship between edge deviation and rolling speed. Thus, we propose a negative feedback control scheme, which formulates this relationship as a linear model on a small scale.

Specifically, to achieve time-efficiency in the edge deviation rectification, we use the linear model to describe the relationship between the rolling speed of spreader rolls and the sheet edge deviation, as shown in (10).

$$\alpha \times f_l(t) - \beta \times f_r(t) = \gamma \times (x(t) - x_s). \tag{10}$$

Here, f_l and f_r denote the frequency of electric motor for the left and right spreader rolls, respectively, which are linearly related to the rolling speed of spreader rolls. $x(t)$ and x_s denote the current edge position and the standard edge position, respectively. α, β, and γ are the ratio factors measured during the production process.

Therefore, the whole control process can be regarded as the combination of a series of linear models with different ratio factors over time, as shown in (10). The ratio factors such as α, β and γ can be different among different linear models over time. We can continuously adjust the parameter γ to dynamically adjust the linear model to approximate the specified relationship. In this way, the relationship between the rolling speed of spreader rolls and the sheet deviation can be depicted in a simplified manner to reduce the deviation rectification complexity.

According to this model, we can adjust the parameters of stretching devices, i.e., f_l and f_r, and rectify the sheet edge to the standard position. However, in the real deployment, these parameters α, β, γ do not only vary among different production lines, but also change with the conveyor speed and the rolling speed of spreader rolls. This leads to the sheet edge drift on the standard position. Moreover, the time delay in the edge position recognition and round-trip feedback further amplifies the jitters of sheet edge alignment. To tackle these issues, we propose the *multi-stage feedback regulation*, which adaptively adjusts the parameters and smooths the results to obtain the stronger delay tolerance and better rectification performance.

In the negative feedback control, we first calculate the smoothed edge position $\widehat{x}(t)$ from the current recognition result and the previous k recognition results, i.e., $x(t-k), \ldots, x(t)$.

Then, we check whether the edge position deviates from the standard range. If not, we maintain the last control parameters f_l and f_r. Otherwise, we start the deviation rectification by changing f_l and f_r to adjust sheet position movement. According to (10), given a sheet position movement $\Delta\widehat{x}(t) = \widehat{x}(t) - \widehat{x}(t-1)$, to minimize the change for control parameters f_l and f_r in both left side and right side, we use the Minimum Mean Square Error (MMSE) method to compute optimal values of $f_l(t)$ and $f_r(t)$ as follows:

$$\arg \min_{f_l(t), f_r(t)} (|f_l(t) - f_l(t-1)| + |f_r(t) - f_r(t-1)|)$$

subject to

$$\gamma \times \Delta\widehat{x}(t) = \alpha \times (f_l(t) - f_l(t-1)) - \beta \times (f_r(t) - f_r(t-1)). \tag{11}$$

Considering that α, β, γ change with the conveyor speed and the rolling speed of spreader rolls from time to time, we need to dynamically update α, β, γ along with time. According to (10), we observe that the value of γ is linear to α and β, so we only need to update γ in an equivalent manner. Therefore, after figuring out the optimal values of $f_l^*(t)$ and $f_r^*(t)$, for the next time slot $t+1$, given the sheet position movement $\Delta x(t+1)$, we can further use the two optimal values to update the parameter γ as follows:

$$\gamma^* = \frac{1}{\Delta\widehat{x}(t+1)}(\alpha \times (f_l^*(t) - f_l(t-1)) - \beta \times (f_r^*(t) - f_r(t-1))). \tag{12}$$

In this way, we can perform the negative feedback control scheme by dynamically adjusting the parameters of the model, while achieving time-efficiency and adaptivity in dynamic environments.

During the process of deviation rectification, when large edge deviation happens, frequent adjustments in rolling spreaders are essentially needed. Inappropriate adjustments will enlarge the jitters of edge position and decrease the rectification performance. To tackle this issue, we propose *Multi-Stage Negative Feedback-based Control* (MSNF) to smooth the frequent adjustment of rolling spreaders. The detailed design of *Multi-Stage Negative Feedback-based Control* (MSNF) is shown in Algorithm 2. When adjusting the parameters of $f_l(t)$ and $f_r(t)$, we use a sliding window $W(t)$ of size k to progressively approach the expected value of $f_l(t)$ and $f_r(t)$ from coarse granularity to fine granularity in a multi-stage manner. Specifically, according to the current edge deviation monitored from *Edge-Eye*, we first calculate the expected change of rotation speed, i.e., $\Delta f_l^*(t)$ and $\Delta f_r^*(t)$, and then divide the change value into k slots in the slide window with the form of index attenuation, e.g., the change values in the 1st, 2nd and following slot are $\frac{\Delta f_l^*(t)}{2}$, $\frac{\Delta f_l^*(t)}{4}$, $\frac{\Delta f_l^*(t)}{8}$ and so on. In this way, the change value can be effectively amortized in the following k slots, i.e.,

$$\Delta f_l^*(t) \approx \sum_{i=t}^{t+k} \Delta f_l^*(t)/2^{i-t+1}$$

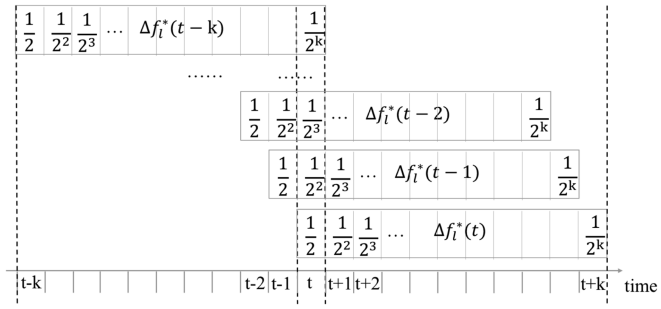


Fig. 15. An example of *Multi-Stage Negative Feedback-based Control*.

$$= \frac{\Delta f_l^*(t)}{2} + \frac{\Delta f_l^*(t)}{4} + \cdots + \frac{\Delta f_l^*(t)}{2^{k+1}}. \tag{13}$$

Moreover, note that during the process of the multi-stage negative feedback-based control, new edge deviation could appear in the slide window $W(t)$ with k slots, due to the issues like nonuniform expansion factors in the IXPE materials. It is essential to further perform rectification on the new edge deviation to progressively approach the expected edge position. To tackle this issue, for current time t, our solution calculates the new edge deviation, i.e., the sheet position movement $\Delta\widehat{x}(t)$, and further divide the difference into the following k slots in the slide window with the form of index attenuation, as shown in *Step 6* in Algorithm 2. Then, for each following slot in the sliding window $W(t)$, it needs to compensate the edge deviation appearing in the past k slots, as shown in *Step 7* in Algorithm 2. In this way, the edge deviation can be progressively rectified in a multi-stage manner, by considering the continuously introduced edge deviations. Fig. 15 shows an example of *Multi-Stage Negative Feedback-based Control*. For each slot t, the edge deviation from the previous k slots should be rectified with specified proportion progressively.

F. Summary

Edge-Eye focuses on providing a deviation rectification system for the real IXPE production, which satisfies high accuracy, time-efficiency and robustness in the limited computing platform. To improve the sensing accuracy in the real scenario, *Perspective Transformation* is used to calibrate the distortion of lens, and *Bi-direction Edge Tracking* is used to detect the edge positions based on the super-resolution image after unsampling. To achieve the time-efficiency, *Reference Bar Selection* and *mROI Extraction* are proposed to reduce the recognition area, and *Frame Caching* is used to skip the recognition of duplicated frames. To achieve the robust performance of deviation rectification, *Negative Feedback Control* with dynamic parameters is proposed to adapt to the dynamic and noisy environments and make the rapid and accurate command response.

V. Performance Evaluation

A. Implementation

Hardware: We have fully implemented *Edge-Eye* and deployed the system in the foaming production line at a local

Algorithm 2: Multi-Stage Negative Feedback-Based Control (MSNF).

Input:
$W(t)$ the sliding window at time t;
$f_l(i), f_r(i), i \in [t-k, t)$: previous rotation frequency in $W(t)$;
$x(i), i \in [t-k, t)$: previous edge position in $W(t)$;
α, β, γ : ratio factors at time t;
Output:
$f_l(t), f_r(t)$: rotation frequency at time t;
for *Each slot t in the slide window $W(t)$* **do**
  Update γ using $f_l^*(t-1)$ and $f_r^*(t-1)$;
  Calculate smoothed edge position $\widehat{x}(t) = \sum_{i=t-k}^{t} x(i)/2^{t-i+1}$;
  Calculate the sheet position movement $\Delta\widehat{x}(t) = \widehat{x}(t) - \widehat{x}(t-1)$, calculate the optimal values $f_l^*(t)$ and $f_r^*(t)$;
  Calculate the expected change of rotation speed $\Delta f_l^*(t) = f_l^*(t) - f_l^*(t-1)$, $\Delta f_r^*(t) = f_r^*(t) - f_r^*(t-1)$;
  For the future k slots in the slide window $W(t)$, for each slot $i \in [t, t+k)$: $\Delta f_l(i) = \Delta f_l^*(t)/2^{i-t+1}$, $\Delta f_r(i) = \Delta f_r^*(t)/2^{i-t+1}$;
  For current time t, update the rotation frequency: $f_l(t) = f_l(t-1) + \sum_{i=1}^{k} \Delta f_l(t-i+1)$, $f_r(t) = f_r(t-1) + \sum_{i=1}^{k} \Delta f_r(t-i+1)$;
  Output $f_l(t), f_r(t)$;
  Move the sliding window forward to $W(t+1)$
end

factory. We describe the system deployment and our evaluation setup and report the performance results from a series of experiments. We use industry-grade alloy as reference bars because the natural color of alloy is highly resistant to high temperature. We deploy two reference bars (i.e., W Bar and B Bar) in the central position inside the vertical heating furnace. A shading plate is placed in front of the recognition area to reduce the light reflection. We deploy a camera (LT-OV2710 of BlueSky Tech) in front of the vertical heating furnace with a distance of 1.2 m from the target material sheet plane. The camera has a wide-angle lens of 120° and its video quality of 1080p/30fps. Thus, one pixel of the image frame captured by this camera represents the width of 1.87 mm on the material sheet plane. We combine the camera with an ARM64 computing platform (Raspberry Pi 4 with 2 GB RAM and 64 GB ROM) in MEN to accomplish the on-device sheet edge detection task. A Programmable Logic Controller (PLC) server receives the recognition result from MEN through Modbus protocol and makes rectification commands for edge deviation. Since the PLC server collects each production line information in 5 Hz, a consistent update frequency contributes to the delivery of control command. Therefore, the time of edge detection needs to be controlled below 200 ms.

Setup: Our IXPE products have 11 different colors, divided into 7 major kinds. For each color, there exist 3 different sizes. For each sheet type, we collect 1 h of production data at the beginning (Init), 2 hours in the middle (Mid), and 1 h in the end (End). With 11×3 different types, we obtain 11×3×4 hours of data in total.

Metrics: For edge detection, we use the recognition accuracy and latency. For deviation rectification, we use the sheet edge alignment error which is highly influenced by the detection performance. For utilization of raw material sheets, we use the ratio of product weight after getting cut and aligned to raw material weight, which depends on the performance of edge alignment.

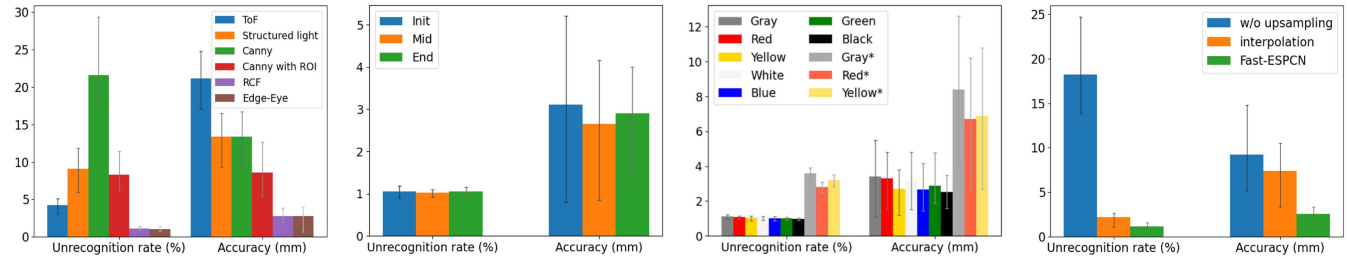
(1) *Edge Position Recognition:* the failure rate refers to the frequency of unsuccessfully detecting the two edges of material sheet during the entire recognition process, and the recognition accuracy rate refers to the difference between the recognition result and the ground truth. (2) *Edge Deviation Rectification:* we define the recognition time delay as the processing time from the frame captured on MEN to the edge position calculated by recognition process, and the control time delay as the time gap between sheet edge appearing in the real production and result received by digital console. Except the recognition task, there exists other tasks, e.g., streaming service, recording service and hardware port communication service. Here we focus on the resource usage of recognition task and ensure MEN can handle all the computing tasks without process scheduling delay. (3) *System Benchmark:* we evaluate the core indicators of IXPE production, the alignment of sheet edge and the utilization rate of raw material sheets. Sheet edge alignment actually reflects the performance of edge deviation rectification, and high-precision alignment result directly improves the utilization rate of raw material sheets. Above metrics directly or indirectly affect the alignment performance. From the barrel effect, it can be seen that a short board can directly lead to the deterioration of alignment, which needs to be avoided as much as possible.

B. Evaluation of Edge Position Recognition

To evaluate the edge position recognition performance, we compare Edge-eye with the distance-based edge detection (ToF, Structured Light), traditional 2D camera-based edge detection (Canny, Canny with ROI) and CNN-based edge detection (RCF). We evaluate the unrecognition rate and the recognition accuracy among different solutions. Specifically, for distance-based edge detection, we respectively use the ToF(Time of Flight)-based camera and structured light-based camera to measure the distance for edge detection. For CNN-based edge detection, we use Richer Convolutional Features (RCF) [20] to perform edge detection, it encapsulates both semantic and fine detail features by leveraging all convolutional features, and achieves state-of-the-art performance on several available datasets.

We also deploy the camera that supports 4 K resolution to evaluate the recognition difference between SR 4 K resolution frames and original 4 K resolution frames.

1) Recognition Rate: In actual production, unavoidable interruption (e.g., material sheet expands or extra rod pulls sheet) may occur, causing the failure of edge detection. We define these cases as unrecognition results when the edge detection method gives no edge position or the recognition result has



(a) Recognition comparison of different solutions (b) Edge-Eye in different production terms (c) Edge-Eye in different material colors (d) Performance evaluation of different upsampling methods

Fig. 16. Evaluation of recognition performance with different settings.

a large distance with the ground-truth, e.g., over 20 mm. By calculating the ratio of unrecognition time to full production time, we evaluate the unrecognition rate for each method.

As shown in Fig. 16(a) and (b), among all solutions, our solution achieves the best performance in recognition rate. The Canny-based solution has the worst performance in recognition rate. *Edge-Eye* achieves an unrecognition rate of 1.04%, and the main unrecognition time is in the initialization stage where material sheet expands and width changes rapidly with extra rod appearance. Since the human worker rectifies the edge deviation during the initialization stage, the unrecognition rate of *Edge-Eye* has no influence on auto rectification performance. Classic Canny operator achieves an unrecognition rate of 21.6% on average, and most of the unrecognition cases occur during the auto rectification stage. Interestingly, when we use Classic Canny operator to process the ROI area of reference bars instead of full frame, the unrecognition rate decreases to 8.3% which shows a high contribution of the reference bars. As shown in Fig. 16(c), we also find the unrecognition rate of the black color material sheet is the lowest among all colors. It comes from the large contrast with the original background, which contributes to the successful recognition of Canny operator with houghlines detection. Also the black color provides the best contrast with W Bar compared with other sheet colors. As the opposite, the gray color material sheet has the highest unrecognition rate because of the low contrast with either W Bar or B Bar.

2) Recognition Accuracy: To evaluate the recognition accuracy, we compare the mean error of the estimated edge position, by measuring the difference between recognition result and the ground-truth of the edge position.

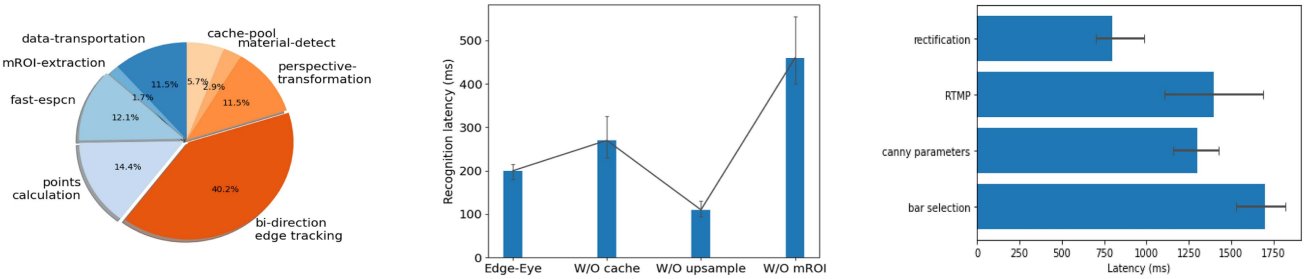
As shown in Fig. 16(a), we find that, among all solutions, our solution achieves the best performance in recognition accuracy, and the ToF-based camera has the worst performance in recognition accuracy. The reason is that, due to the presence of glass between the camera and the material, as well as the unstable lighting condition, the depth information cannot be accurately measured, thus it is not suitable to be used as an effective metric for edge detection. For the CNN-based edge detection, RCF achieves very close performance to our solution, however, it incurs large compute complexity and brings high latency in the edge detection process. Thus it is not suitable to support real-time edge deviation rectification.

We further evaluate the recognition performance of Edge-Eye in different production terms and different colors, as shown in Fig. 16(b) and (c). We find that, if without reference bars, *Edge-Eye* will achieve the lower recognition accuracy, as illustrated by Gray*, Red* and Yellow* in Fig. 16(c). That is, the contrast between the sheet and the background is reduced in the absence of reference bars, which would greatly affect the performance of edge detection. Judging from the production terms in Fig. 16(b), the position error at the beginning stage is relatively large, with an average position error of 3.6 mm. This is because that the material edge will get bent when the material expands in the initial stage, such that the issue of edge blur appears and the error of edge detection increases.

To evaluate the accuracy improvement from image upsampling, we evaluate the performance of no upsampling, upsampling with interpolation as well as upsampling with Edge-Eye, as shown in Fig. 16(d). We find that, for no upsampling method, both the recognition rate and the recognition accuracy is very low. For upsampling with interpolation, it can effectively improve the recognized rate, but the accuracy is still not high enough to support millimeter-level edge recognition. For upsampling with Edge-Eye, it achieves both high recognition rate and recognition accuracy, and it greatly outperforms the other two solutions.

3) Latency Analysis: During the production stage, *Edge-Eye* achieves an average recognition time of 200 ms in edge detection. As shown in Fig. 17(a), the module of bi-direction edge tracking consumes the most time. Besides, the cache module has to calculate the image entropy for each frame in the cache pool, thus it consumes extra time. In Fig. 17(b), we evaluate the recognition latency after disabling the module of cache, image upsampling, and mROI extraction in succession. Without module of cache, the average recognition time increases to 270 ms. Without module of image upsampling, the average recognition time decreases to 110 ms. Without module of mROI extraction, *Edge-Eye* directly tracks edge position in entire reference bar area and achieves an average latency of 460 ms. It can be clearly found that the mROI extraction module reduces the latency most, and the module of image upsampling increases the extra latency, but without this module, the position error increases from 2.8 mm to 5.4 mm. From the perspective of material production, this is a considerable trade-off between latency and accuracy.

We further evaluate the latency in initialization, messaging and rectification, which are not included in the edge detection process. As shown in Fig. 17(c), for the initialization latency, we evaluate the latency of reference bar selection and Canny

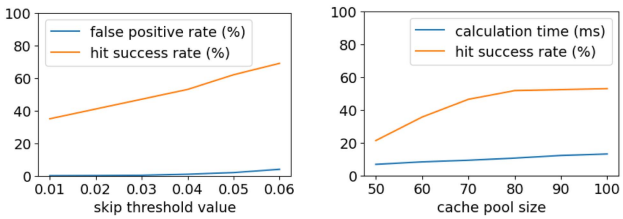


(a) Latency analysis for edge detection process

(b) Latency comparison without specific modules

(c) Latency analysis for initialization, messaging and rectification

Fig. 17. Latency analysis.



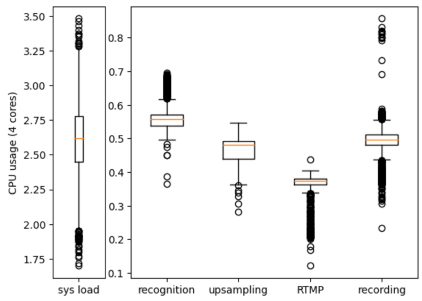
(a) Analysis of skip threshold

(b) Analysis of pool size

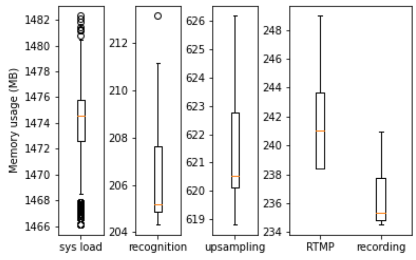
Fig. 18. Evaluation of hyper-parameters of cache pool.

operator parameters generation. The experiment results show that our solution achieves average latency of 1,700 ms and 1,300 ms for reference bar selection and Canny operator parameters generation, respectively. The latency is fairly small and it can be effectively amortized by the following edge detection and rectification phase. For the messaging latency, we evaluate the latency of RTMP media service, it represents the time interval between the appearance of a sheet in the detection area and the corresponding frame appearing in the central control room. Our solution achieves an average latency of 1,400 ms for RTMP media service with limited computing resources in MEN. For the rectification latency, we evaluate the overall latency of edge deviation rectification, including the process of edge detection and edge deviation rectification. Our solution achieves an average latency of 800 ms for rectification latency. Considering that the conventional rolling speed is about 1 m/s, this latency is qualified for providing reliable performance in edge deviation rectification.

4) Cache Pool Analysis: We further evaluate the performance of cache pool scheme through hit success rate, hit calculation time and false positive rate, by adjusting the skip threshold and cache pool size. The hit success rate refers to the probability of successfully finding duplicated frames, and the false positive rate refers to the probability of mistaking the current frame as the duplicated one. 1) *Skip Threshold:* We make further experiments to determine the optimal skip threshold for cache pool, and the result is shown in Fig. 18(a). As the skip threshold increases, the hit success rate increases. However, when the skip threshold is greater than 0.03, the false positive rate increases rapidly, affecting the further recognition accuracy. In fact, the entropy value represents the similarity between the two different frames in the pixel level. Thus, the larger skip threshold will lead to the



(a) Box analysis for CPU usage in MEN



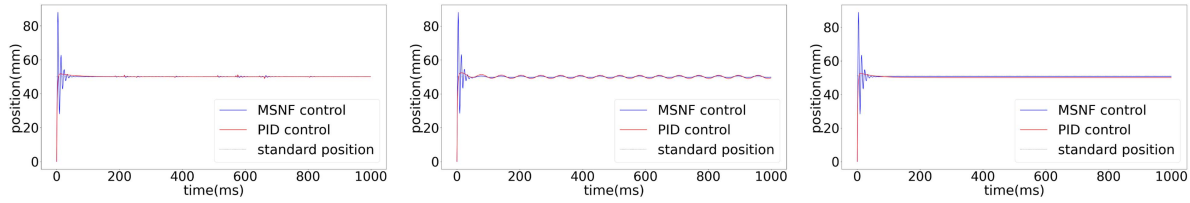
(b) Box analysis for memory usage in MEN

Fig. 19. Resource consumption in mobile edge node.

wrong hit for the frames without enough similarity. Finally, we set the skip threshold to 0.03, which can reduce the recognition time without degrading the recognition accuracy. 2) *Cache Pool Size:* A large size of cache pool always leads to higher hit success rate, but also enlarges the hit calculation time. Here we conduct experiments about cache pool size for best recognition performance. As shown in Fig. 18(b), the calculation time increases linearly with pool size, but the hit success rate increases slowly after the pool size exceeds 80. The slow increase of hit success rate means that most of similar frames are cached in the pool, thus a larger pool size contributes little to the hit success rate but greatly increases the hit computation time. Therefore, we choose the value of 80 as our cache pool size.

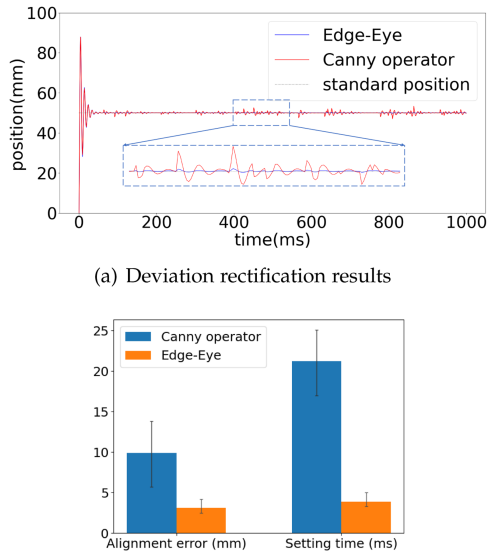
C. Resource Consumption in Mobile Edge Node

1) CPU Usage in Mobile Edge Node: To keep the Mobile Edge Node (MEN) run steadily in a long-term manner, we need to ensure all tasks run fluently without long wait time on process



(a) Gaussian noise ($\mu = 0, \sigma = 1$) (b) Sinusoidal noise ($A = 1, \omega = 0.1$) (c) Step noise ($A = 1$)

Fig. 20. Edge deviation rectification performance in different conditions.



(a) Deviation rectification results

(b) Alignment error and setting time

Fig. 21. Performance on deviation rectification with Edge-Eye and fine-tuned Canny operator.

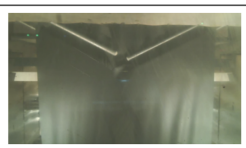
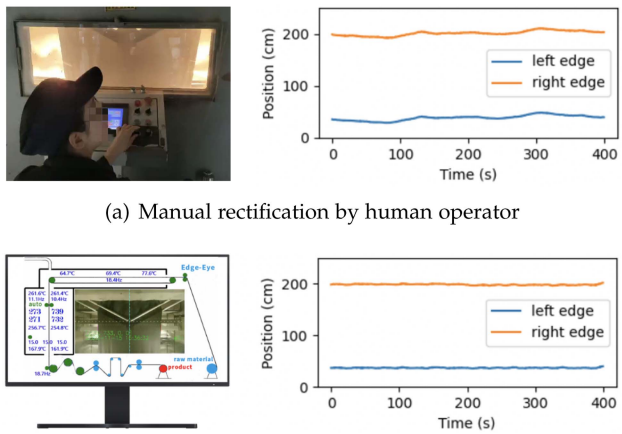
Case	Photo	Accuracy
Light reflection		6.4mm
Weak background light		4.8mm
Glass dirt		2.83mm

Fig. 22. Some typical challenging conditions.



(a) Manual rectification by human operator

(b) Automatic rectification by Edge-Eye

Fig. 23. Comparison of different rectification methods.

scheduling. We thus look into the resource consumption for tasks including edge recognition, frame capturing, RTMP streaming, video recording and Modbus communication. We evaluate the average and max-min value to ensure the MEN can run in long term with reasonable load in the edge detection stage. As shown in Fig. 19(a), we can clearly find a total of 75% of computing resources on MEN are used. Among them, the recognition task uses 8.75%-17.5% of the overall CPU resources, the upsampling task uses 7%-13.5% of the overall CPU resources, the RTMP task uses 3.5%-11% of the overall CPU resources, the recording task uses 6%-22% of the overall CPU resources. In all, *EdgeEye* can run steadily without incurring heavy computing load.

2) Memory Usage in Mobile Edge Node: We analyze the memory usage for the corresponding tasks. With loss of generality, the overall memory in our setting is 2,048 MB in total. As shown in Fig. 19(b), it is found that, in the edge detection stage, the total memory usage is 1,475 MB on average. Among them, the recognition task uses 10.3% of the overall memory resources, the upsampling task uses 30% of the overall memory resources, the RTMP task uses 11.7% of the overall memory resources, the recording task uses 11.3% of the overall memory resources. In all, the total memory usage for the whole system does not exceed the memory limitation, which means *Edge-Eye* can run steadily without incurring heavy memory usage.

D. Evaluation of Edge Deviation Rectification

1) PID Controller versus MSNF: We evaluate the rectification performance for sheet edge deviation on the actual production line. We implemented a standard rectification approach with PID controller. We compare the performance of edge deviation rectification between PID controller and our method

MSNF, i.e., the multi-stage negative feedback control. Specifically, we perform the evaluation under different conditions of environmental noises in production lines, i.e., the edge deviation varies with time in different patterns, which are Gaussian noise, Sinusoidal noise, and Step noise, respectively. Fig. 20 show the rectification performance of PID controller and MSNF in the above conditions. As shown in Fig. 20(a), in the condition of Gaussian noise, we built a well-tuned PID controller according to the training data with Gaussian distribution. We can find that the PID controller does have lower mean error than MSNF, i.e., 2 mm versus 4.2 mm, moreover, the PID controller has close convergence time with MSNF, i.e., 0.12 s versus 0.14 s. The reason of PID controller outperforming MSNF is that, the PID controller was well tuned with a large amount of training data with the prior knowledge of Gaussian distribution.

However, in the real production scenarios, different production lines, or even the same production line with different types of IXPE, have different production conditions, such as production speed, production temperature and material thickness. Hence, a well-tuned PID controller for one production line may suffer a sharp drop in performance on another production line. Moreover, the specific training cost for each production line one by one is too high to be acceptable. Therefore, we further evaluate the performance in different noise distributions. As shown in Fig. 20(b), in the condition of Sinusoidal noise, the mean error of MSNF is 6.2 mm, which is better than 7.6 mm for PID controller. As shown in Fig. 20(c), in the condition of Step noise, the mean error of MSNF is 4.3 mm, which is better than 5.7 mm for PID controller. The reason is that, the PID controller requires well-tuned parameters according to the exact variation patterns. Without the exact prior knowledge, the PID controller cannot perform well. In contrast, MSNF can automatically adjust the parameters and adapt to different conditions, so it outperforms the PID controller in most conditions.

2) The Impact From Different Edge Deviation Detection Schemes: To evaluate how edge deviation detection accuracy affects deviation rectification accuracy, we perform the multi-stage negative feedback control based on two different edge deviation detection schemes, i.e., the edge detection scheme of Edge-Eye and the fine-tuned Canny operator. We find that, with different edge deviation detection schemes, the performance of deviation rectification varies a lot. As shown in Fig. 21(a), since the Canny operator cannot provide accurate edge estimation, there exist nonnegligible fluctuations for the rectified edge position over time. As a contrast, with our Edge-eye-based solution, the edge position is rectified very smoothly over time. We further evaluate the edge alignment error and the setting time. Here, the setting time refers to the time interval required for the edge to reach the range of the allowable error after the edge deviation happens. As shown in Fig. 21(b), with the Canny operator, the system achieves average alignment error of 9.9 mm and average setting time of 21.2 ms, in comparison, with Edge-Eye-based solution, the system achieves average alignment error of 3.1 mm and average setting time of 3.9 ms. This implies that the edge deviation detection accuracy indeed affects deviation rectification accuracy. Moreover, Edge-Eye-based rectification outperforms the Canny operator-based rectification in both alignment error and convergence speed.

E. Case Study

Edge-Eye has been actually deployed in a large IXPE manufacturing enterprise. Specifically, 36 production lines are equipped with *Edge-Eye* system for automatic edge deviation rectification over 20 months. Here, in addition to the above experiment results in real production, we show some results towards some typical challenging conditions in Fig. 22.

1) *Light reflection:* The deployment of shading plate reduces most of the light reflection, but still in some cases the light reflects to key recognition area and causes huge interference. When light reflects to the mROI area, it makes the vertical edge tracking unreliable, thus decreases the average recognition accuracy to 6.4 mm. Due to the calculation of background edge points, the reflection light will be considered as the background and get subtracted from sheet edge points. In this way, the reflection light will cause the slight effect on the deviation rectification. Further design of the shading plate will be required to improve the stability of recognition.

2) *Weak background light:* It comes from the light equipment malfunction in the vertical heating furnace and lasts for a short time. In this case, the camera captures dark frames and reference bars provide less contrast with sheet color. Similar with the issue of light reflection, the vertical edge tracking fails to work, and the horizontal edge tracking may experience an accuracy drop. Nevertheless, we can achieve the average recognition accuracy of 4.8 mm, which is acceptable for automatic rectification.

3) *Glass dirt:* As the small dirt is treated as the common background edge point, it brings no effect on the edge detection. In this case, the average recognition accuracy is 2.83 mm, which is consistent with the accuracy of normal situation. The result indicates that *Edge-Eye* can achieve the millimeter-level accuracy for edge recognition even in challenging conditions, satisfying the requirements of actual production scenarios.

Meanwhile, we compare the variation of edge position between the traditional human supervision and our solution *Edge-Eye*. As shown in Fig. 23, the edge position with the automatic rectification by *Edge-Eye* is much smoother than the manual rectification. In general, during the process of 20-month real deployment, 66 manpower per day (90% of the overall manpower) has been saved for 36 production lines, the utilization rate of IXPE material increases from 87% to 94%, thus the comprehensive output value has increased 400,000 dollars per month.

VI. Conclusion

In this paper, we propose *Edge-Eye*, a camera-enabled automatic edge deviation rectification system for IXPE production with mm-level accuracy. To achieve the *robust* edge detection, we deploy a pair of high contrast reference bars to enhance the edge contrast. To achieve the *time-efficient* edge detection with high *accuracy* in MEN, we use the minimized ROI extraction and cache method to reduce computing resources in both space and time domains, and then adopt the image upsampling method to improve the frame resolution for bi-direction edge tracking. Finally, *Edge-Eye* automatically rectifies the edge deviation in fine-grained level using multi-stage negative feedback control.

We implemented *Edge-Eye* on the ARM64 platform. The real evaluation of 20-month deployment for 36 production lines shows that 90% of manpower is saved, and the utilization rate of IXPE material increases from 87% to 94%.

Lei Xie and Chuyu Wang are co-corresponding authors.

REFERENCES

[1] H. Wang, F.-J. Zhao, and Y.-K. Deng, "Development and application of the millimeter wave SAR," *J. Infrared Millimeter Waves*, vol. 34, no. 4, pp. 452–459, 2015.

[2] J. Shan and C. K. Toth, *Topographic Laser Ranging and Scanning: Principles and Processing*. Boca Raton, FL, USA: CRC, 2018.

[3] J. S. Seybold, *Introduction to RF Propagation*. Hoboken, NJ, USA: Wiley, 2005.

[4] C. Hu, L. Dai, T. Mir, Z. Gao, and J. Fang, "Super-resolution channel estimation for mmWave massive MIMO with hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8954–8958, Sep. 2018.

[5] X. Chen, J. Xi, Y. Jin, and J. Sun, "Accurate calibration for a camera–projector measurement system based on structured light projection," *Opt. Lasers Eng.*, vol. 47, no. 3/4, pp. 310–319, 2009.

[6] M. Kytö, M. Nuutinen, and P. Oittinen, "Method for measuring stereo camera depth accuracy based on stereoscopic vision," in *Three-Dimensional Imaging, Interaction, and Measurement*, vol. 7864. Bellingham, WA USA: International Society for Optics and Photonics, 2011, Art. no. 78640I.

[7] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London. Ser. B. Biol. Sci.*, vol. 207, no. 1167, pp. 187–217, 1980.

[8] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, Apr. 1988.

[9] W. Dong and Z. Shisheng, "Color image recognition method based on the prewitt operator," in *Proc. IEEE Int. Conf. Comput. Sci. Softw. Eng.*, 2008, vol. 6, pp. 170–173.

[10] A. Rosenfeld, "The max roberts operator is a hueckel-type edge detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 1, pp. 101–103, Jan. 1981.

[11] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[12] G. Bertasius, J. Shi, and L. Torresani, "DeepEdge: A multi-scale bifurcated deep network for top-down contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4380–4389.

[13] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[14] M. Maire, S. X. Yu, and P. Perona, "Reconstructive sparse code transfer for contour detection and semantic labeling," in *Proc. Asian Conf. Comput. Vis.*, Springer, 2014, pp. 273–287.

[15] Z. Yu, C. Feng, M. Liu, and S. Ramalingam, "CASENet: Deep category-aware semantic edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5964–5973.

[16] X. S. Poma, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust CNN model for edge detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1923–1932.

[17] P. Cohen, J. Weng, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 10, pp. 965–980, Oct. 1992.

[18] E. J. O'neil, P. E. O'neil, and G. Weikum, "The LRU-K page replacement algorithm for databasedisk buffering," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 297–306, 1993.

[19] Z. Chu et al., "A generalizable sample resolution augmentation method for mechanical fault diagnosis based on ESPCN," *J. Sensors*, vol. 2021, pp. 1–11, 2021.

[20] Y. Liu et al., "Richer convolutional features for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1939–1946, Aug. 2019.

Lei Xie (Member, IEEE) received the BS and PhD degrees from Nanjing University, China, in 2004 and 2010, respectively, all in computer science. He is currently a professor with the Department of Computer Science and Technology, Nanjing University. He has published more than 100 papers in *IEEE Transactions on Mobile Computing*, *ACM/IEEE Transactions on Networking*, *ACM Transactions on Sensor Networks*, ACM MobiCom, ACM UbiComp, ACM MobiHoc, IEEE INFOCOM, IEEE ICNP, IEEE ICDCS, etc.

Zihao Chu received the BS degree from the School of Software, Dalian University of Technology, China, in 2019. He is currently working toward the master's degree with the Department of Computer Science and Technology, Nanjing University, China. His research interests include Industrial IoT and computer vision.

Yi Li received the BS degree from the School of Computer Science and Technology, Shandong University, China, in 2020. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Nanjing University, China. His research interests include multimodal sensing and sensing intelligence.

Tao Gu (Member, IEEE) received the BEng degree in automatic control from the Huazhong University of Science and Technology, the MSc degree in electrical and electronic engineering from Nanyang Technological University, and the PhD degree in computer science from the National University of Singapore. He is currently a professor with the School of Computing, Macquarie University, Sydney. His research interests include Internet of Things, embedded AI, mobile computing, ubiquitous computing, and Big Data analytics. His publications usually appear in journals and conferences, including *IEEE/ACM Transactions on Networking*, *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Mobile Computing*, MobiCom, SenSys, UbiComp, IPSN, and INFOCOM.

Yanling Bu (Member, IEEE) received the PhD degree in computer science from Nanjing University, China, in 2022. She is currently an associate researcher with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. Her research interests include Internet of Things (IoT), mobile sensing, and RFID systems.

Chuyu Wang (Member, IEEE) received the PhD degree in computer science from Nanjing University, China, in 2018. He is currently an associate researcher with the Department of Computer Science and Technology, Nanjing University. His research interests include RFID systems, software-defined radio, activity sensing, indoor localization, etc.

Sanglu Lu (Member, IEEE) received the BS, MS, and PhD degrees from Nanjing University, China, in 1992, 1995 and 1997, respectively, all in computer science. She is currently a professor with the Department of Computer Science and Technology, Nanjing University. Her research interests include distributed computing and pervasive computing.