

Article

A Novel Relocalization Method-Based Dynamic Steel Billet Flaw Detection and Marking System

Hongxing Zhou , Juan Chen, Qinghan Hu , Xue Zhao and Zhiqing Li *

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; hongxingzhou@mail.buct.edu.cn (H.Z.); jchen@mail.buct.edu.cn (J.C.); 2022400203@mail.buct.edu.cn (Q.H.); 2021210506@mail.buct.edu.cn (X.Z.)

* Correspondence: lizhiqing@mail.buct.edu.cn

Abstract: In the current steel production process, occasional flaws within the billet are somewhat inevitable. Overlooking these flaws can compromise the quality of the resulting steel products. To address and mark these flaws for further handling, Magnetic Particle Testing (MT) in conjunction with machine vision is commonly utilized. This method identifies flaws on the billet's surface and subsequently marks them via a device, eliminating the need for manual intervention. However, certain processes, such as magnetic particle cleaning, require substantial spacing between the vision system and the marking device. This extended distance can lead to shifts in the billet position, thereby potentially affecting the precision of flaw marking. In response to this challenge, we developed a detection-marking system consisting of 2D cameras, a manipulator, and an integrated 3D camera to accurately pinpoint the flaw's location. Importantly, this system can be integrated into active production lines without causing disruptions. Experimental assessments on dynamic billets substantiated the system's efficacy and feasibility.

Keywords: machine vision; steel billet; descriptor matching; point cloud; SVD (singular value decomposition)



Citation: Zhou, H.; Chen, J.; Hu, Q.; Zhao, X.; Li, Z. A Novel Relocalization Method-Based Dynamic Steel Billet Flaw Detection and Marking System. *Electronics* **2023**, *12*, 4863. <https://doi.org/10.3390/electronics12234863>

Academic Editor: Dah-Jye Lee

Received: 29 September 2023

Revised: 11 November 2023

Accepted: 28 November 2023

Published: 2 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In industrial steel production, occasional slender flaws on the steel surface, which are challenging to discern with the naked eye, might precipitate crack expansion post rolling. This can have direct ramifications for the subsequent phases of steel production. Given the implications, it is imperative to detect and rectify surface flaws. In many contemporary steel mills, flaws are manually pinpointed and chalk-marked, polished in a subsequent procedure. Given this, the development of an automated flaw detection-marking system becomes essential to optimize labor costs and enhance operational efficiency.

Several methodologies exist for detecting flaws on billet surfaces, including eddy current testing, infrared testing, and magnetic particle testing (MT) [1]. Among these, MT stands out due to its reliability, efficiency, and non-damaging nature to billets [2]. Under specific lighting conditions, the magnetic particle reaction facilitates easy visual identification of flaws. Numerous studies have delved into the utility of machine vision for steel billet surface detection. A typical visual tool offers dual functionalities: flaw detection and classification [3–8]. Upon flaw localization on the billet surface, the positional data must be relayed to the marking device, such as a manipulator [9].

Generally, an interlude of magnetic particle cleaning is executed between MT and marking. In the manufacturing landscape, this cleaning spans an extended length of the billet. Given potential discrepancies in roller table installations, the billet's orientation may shift during this extended process, potentially leading to misalignment between the marking device and the flaw's position. To address this, a proximal visual system precedes the manipulator for flaw relocalization. Our proposed solution leverages point clouds rendered by a singular 3D camera.

For mobile workpieces, as described in work [10], a line scanning camera inspects rapidly moving tubular metals, identifying flaw locations. However, billet production systems might necessitate multiple line scanning cameras, thereby inflating costs. Owing to its expansive illumination range coupled with cost effectiveness, we integrated a solitary area-array camera into the system. This paper offers the following contributions:

- A novel system for locating and marking billet flaws that can be operated without affecting billet production.
- A dynamic keypoint tracking strategy combining descriptor extraction and a projection technique. Descriptor extraction employs a SIFT (Scale Invariant Feature Transform)-like method, whereas the projection technique utilizes geometric constraints and point cloud image processing algorithms. SIFT is an algorithm for detecting and describing key points in an image with stability for rotation, scaling, brightness changes, etc.
- Implementation of an SVD-based technique for the ICP (Iterative Closest Point) solve, facilitating the calibration of both the camera and manipulator.

2. Related Work

Theoretically, the pose estimation of the billet involves edge detection and the Hough transform to discern the billet's edge, subsequently computing its pose transformation. Notably, edge detection for deducing billet poses remains underexplored, with predominant focus directed towards flaw detection. Liu et al. [11] deployed the Sobel method for steel surface defect detection, achieving a detection rate of 80%. For augmenting the precision of heavy rail surface defect detection, an enhanced Sobel operator was introduced in [12]. By incorporating a template oriented at 45° and 135°, this operator captured a more comprehensive edge information compared to its traditional counterpart. Another study [13] utilized the Canny edge extraction operator to extract steel strip image edges and delineate its contour. Our endeavors with edge detection techniques are elaborated upon in Section 3.

There is also feature extraction from point clouds of billets to achieve their positional detection. In [14], a multi-step refinement method used robust moving least squares to fit potential features locally. But finer calculations inevitably bring a heavier computing burden. In [15], a multi-scale neighborhood feature extraction and aggregation model (MNFEAM) to enhance feature extraction method for point cloud learning was proposed. Normally, deep learning-based methods require a large enough number of training samples, demanding significant time costs.

Furthermore, billet pose identification can be realized via template matching [16]. However, the accuracy of this approach might be compromised due to variances in surface textures and morphologies across different billets. Acknowledging this, our methodology incorporates the region growth algorithm in tandem with RANSAC (Random Sample Consensus) for implementation.

Building upon the existing methods for flaw detection in steel billets, this study aims to address the practical challenges of integrating such systems into the dynamic environment of industrial steel production. Our research focuses on developing an automated system that not only detects surface flaws with high accuracy, but also aligns flaw marking processes with real-time orientation changes of billets. This objective stems from the identified need for more operationally feasible solutions in the existing literature.

3. Application Scenario Introduction and Previous Attempts

3.1. Application Scenario Introduction

The proposed scheme is implemented within a steel production line. Here are specific details pertaining to the billets processed on this line.

Figure 1 presents an illustration of a billet. Typically, the flaws manifest as straight lines parallel to the direction of forward movement, with lengths ranging from 50 mm to 500 mm. The spacing between roller tables is approximately 2 m. Billets on this line measure between 13 m to 15 m in length and operate at a speed of 0.2 m/s. Due to the manufacturing

methodology, flaws exclusively appear on the two inclined planes at the upper part of the billet. A section of roughly 8 m is designated for magnetic particle cleaning.

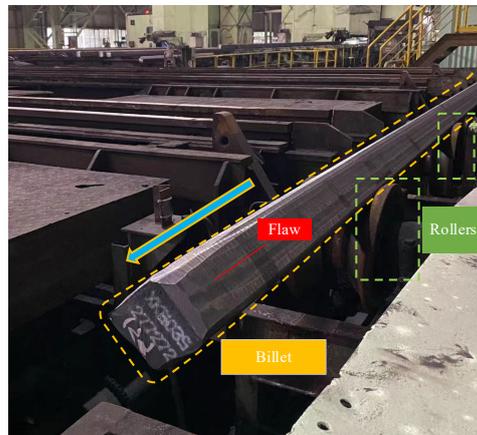


Figure 1. Shape of a billet in the production workshop. The direction of the arrow indicates the travel direction of the billet.

In reality, our system is not constrained by the length of the steel billets; it can be readily adapted to billets of different cross-sectional areas with minor adjustments to the installation of the flaw detection cameras. For slab billets, which can be considered as a type of billets with a larger ratio of width to thickness, our approach remains effective. This adaptability ensures that the system is versatile and applicable across a range of billet dimensions.

In essence, the object of the proposed system is to relocalize flaws running parallel to the direction of movement on the two upper surfaces of the billet after it travels a considerable distance. Following this, their positional information is translated into the manipulator's coordinates to facilitate marking.

3.2. Previous Attempts

Initially, our relocalization was executed using two well-calibrated cameras, similar to the flaw detection setup. Each camera was equipped with supplementary lighting, as shown in Figure 2a,b. By fine-tuning the parameters of the cameras and the lightings, we enhanced the visibility of the edges of the steel billets under the illumination of the supplementary lights, as depicted in Figure 2c. The workflow for this process is as follows:

- Midpoints are selected on the upper edges of the images captured by both cameras, along with two equidistant points on either side, serving as co-visibility points.
- These co-visibility points are then triangulated and processed using least squares estimation to reconstruct the spatial position of a co-visibility line.
- Utilizing the known actual width of the steel billet and the minimum width difference observed on the image's upper and lower edges, the spatial position of the lower edge is deduced.
- Utilizing the spatial positions of the upper and lower edges, the position of the plane is reconstructed, thereby facilitating the calculation of the flaw's location.

The viability of this approach is highly contingent on specific lighting conditions and installation requirements. In real-world production environments, variables such as changing illumination and vibrations can impact the precision of this method. Therefore, the robustness of this solution is somewhat limited under these conditions. As a result, this approach was deemed unviable and subsequently discarded. The workflow of the validated, feasible solution is depicted in Figure 3.

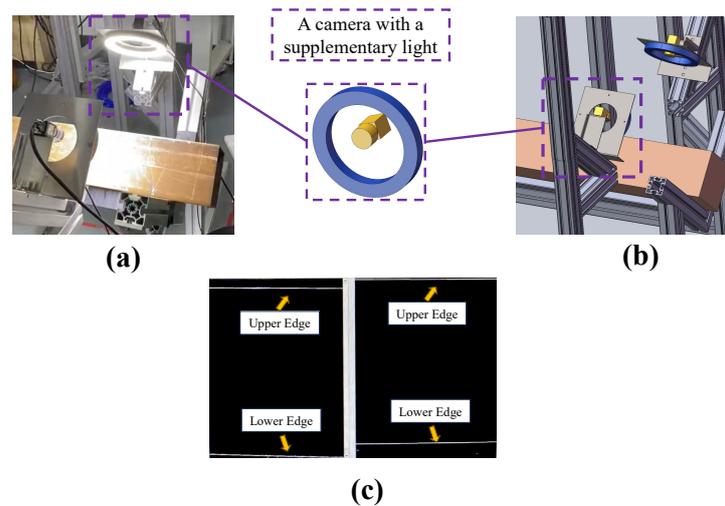


Figure 2. Previous solution. (a) The previous solution. (b) Design diagram of the previous solution. (c) Billet edge detection in the previous solution.

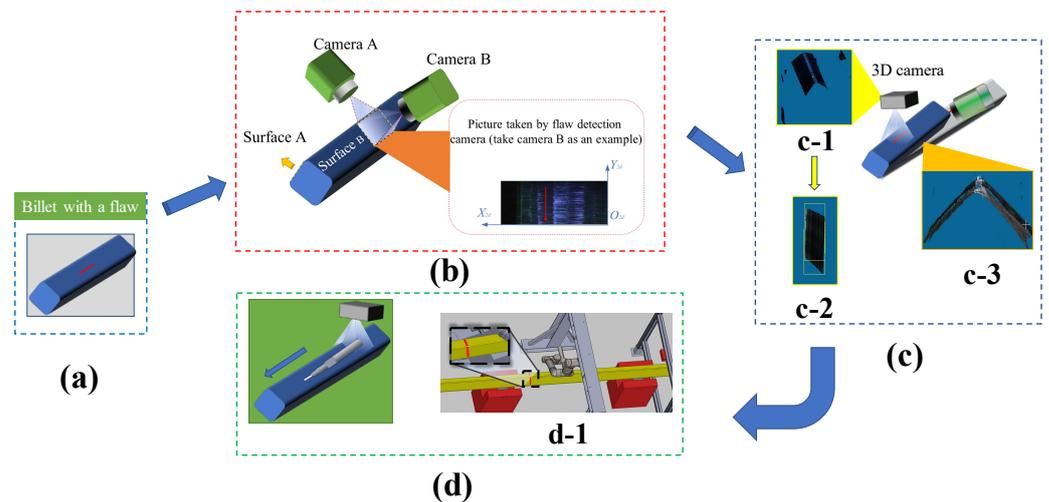


Figure 3. Workflow. (a) Billet with a flaw; (b) The processing of MT flaw detection; (c) In view of posture differences of a billet, a 3D camera is brought in for relocalization, where (c-1) represents the original point cloud data of the billet, (c-2) shows the effect after processing, and (c-3) is to illustrate that the posture of the same position on the same billet changes when passing by the fixed 3D camera multiple times; (d) The processing of the coordinates conversion system, where in (d-1), the red line demonstrates the working range of the chalk on the manipulator. Finally, the manipulator clamps a piece of chalk to complete the marking action.

4. Detailed Architecture

The system encompasses two primary components: (1) a flaw detection system, equipped with two 2D industrial cameras, and (2) a marking system, which includes a 3D camera paired with a manipulator. A host computer orchestrates the operations, computations, and data communications of the entire system.

4.1. Flaw Detection System and Notations

4.1.1. Flaw Detection

As depicted in Figure 3b, the two flaw detection cameras are designed to capture images of the two surfaces of the steel billet where flaws are likely to manifest. Each camera is meticulously positioned to ensure that its field of view encompasses the entire width of the billet. As illustrated, the image captured by Camera B has its x -direction correlated with the width of the surface. The flaw detection procedure identifies which surface (either

A or B) presents a flaw. The flaw’s starting position along the x -direction is discerned using a recognition algorithm. For the y -direction coordinate of the flaw’s initiation point, both sections utilize identical encoders (refer to Figure 4) that are engaged with the moving billet. This mechanism facilitates the computation of the coordinate. The same method is employed to determine the flaw’s length. As a result, the visual system design negates the need for considerations related to the y coordinates.

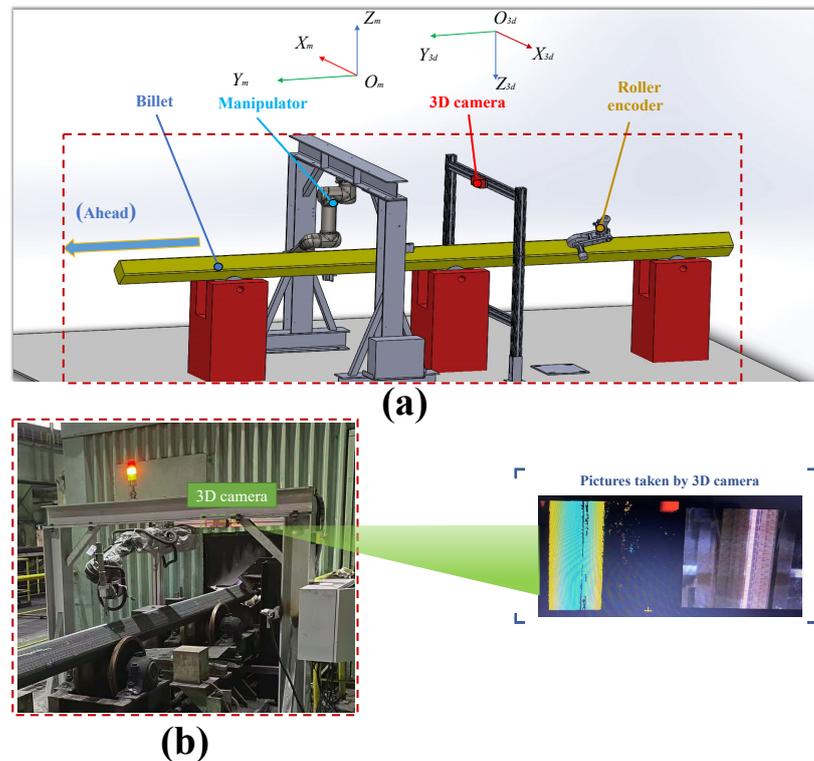


Figure 4. Marking system. (a) Engineering design drawing (SolidWorks). (b) General site layout.

4.1.2. Notations

Figure 4 offers a schematic representation of the marking system’s structure. At the top of Figure 4a, the coordinate systems for both the 3D camera and the manipulator are illustrated, represented by $X_{3d} - Y_{3d} - Z_{3d}$ and $X_m - Y_m - Z_m$. Y_{3d} and Y_m , respectively. Notably, both Y_{3d} and Y_m align with the billet’s direction of movement.

4.2. Relocalization

The 3D camera, positioned above the steel billet, is adept at producing both RGB and point cloud images. As indicated in Figure 3(c-3), vibrations lead to discrepancies when the same billet position moves under the 3D camera, causing point cloud images not to overlap. Hence, relocalization becomes indispensable.

4.2.1. Descriptor-Based Relocalization

The flaw’s starting point is identified by one of the detection cameras. As it transitions towards the 3D camera, it might undergo unforeseen displacements due to vibrations. Consequently, relocalizing the flaw is tantamount to tracking a dynamic point. However, this presents a twofold challenge: 1. The considerable separation between the two cameras precludes either from maintaining a consistent track of the point. 2. Potential variations stemming from the magnetic particle cleaning process, coupled with fluctuating lighting conditions, can modify the point’s features, impeding consistent visual tracking. To address these challenges, the proposed system leverages encoders to relay the point’s position to the 3D camera.

$$P_{flaw-3D} = T_{e,3D}(T_{2D,e}P_{flaw-2D} + f(v)), \tag{1}$$

where $P_{flaw-3D}$ and $P_{flaw-2D}$ represent the coordinates of the detected flaw's starting point in the 3D and 2D cameras, respectively. $T_{e,3D}$ and $T_{2D,e}$ are transformation matrices in a SE(3) form. These matrices capture the transformations from the encoder to the 3D camera and from the 2D camera to the encoder, respectively. Function $f(v)$ characterizes the offset induced by vibrations.

Given the inability to track feature points, the starting point of the flaw is treated as a keypoint. Consequently, texture descriptors similar to SIFT are extracted. SIFT has demonstrated robustness to variations in lighting, changes in scale, and rotations. We consider the starting point of a flaw as a center point and extract a descriptor from this one. We term this method the SIFT-like algorithm. The representation for extracting SIFT-like descriptors is provided below. We suppose the pixels around the keypoint (the starting point of the flaw) are partitioned into n subdivisions. The gradient magnitude and gradient direction for pixels within the i th subdivision are symbolized as g_i and θ_i , respectively. The descriptor of the keypoint can be represented as an $n - dimensional$ vector $D = \{d_1, d_2, \dots, d_n\}$, where

$$d_i = \left(\sum_{j=1}^m g_{i,j} \cos(m\theta_{i,j}), \sum_{j=1}^m g_{i,j} \sin(m\theta_{i,j}) \right). \tag{2}$$

The positive integer m denotes the number of sub-vectors present within each region. The host computer continuously tracks encoder values. Upon detecting a keypoint proximate to the 3D camera, the system derives descriptors for the prevailing image. The matching SIFT-like algorithm is given by

$$dist(P_{flaw-3d} - P_{flaw-2d}) = \|D_{i3d} - D_{i2d}\|. \tag{3}$$

here, $\|\cdot\|$ signifies the Euclidean norm of a vector. If the distance, $dist$, between two descriptors falls below the threshold h_1 , they are deemed to match, and the point is recognized as a keypoint.

4.2.2. Projection-Based Relocalization

If there is a pronounced rotation, the precision of SIFT diminishes substantially. In such scenarios, adopting a more resilient strategy based on geometric constraints becomes beneficial. Projecting these constraints onto the 3D camera facilitates deriving the coordinates of specific points. For illustrative purposes, let us consider the flaw on surface B. The method to translate its coordinate from camera B to the 3D camera is depicted in Figure 5.

From Figure 5a:

$P(x_{2d}, y_{2d})$ —the starting point of the flaw,

L —the width of the billet, generally 140 mm.

In Figure 5b:

α —the plane $y_{3d} = 0$,

$P'(x'_{3d}, 0, z'_{3d})$ —the intersection of plane α and the flaw. It is the same as the x_{3d} and z_{3d} coordinates of the starting point of the flaw.

P'_{xmin} —the point with the lowest x coordinate x_{min} on the intersection of plane α and surface B,

P'_{xmax} —the point with the largest x coordinate x_{max} on the intersection of plane α and surface B,

L' —difference between x coordinates of P_{xmax} and P_{xmin} .

As can be seen from the figure,

$$\frac{x_{2d}}{L} = \frac{x_{max} - x'_{3d}}{L'}. \tag{4}$$

Thus,

$$x'_{3d} = x_{max} - \frac{x_{2d}L'}{L}. \tag{5}$$

Coordinate x'_{3d} can thus be determined. For simplification, we directly consider the y -coordinate as zero. When the x_{3d} and y_{3d} coordinates of a point within a generic point cloud image are known, the corresponding z_{3d} coordinate is derivable.

Let us assume that the positions of point $P_{flaw-3d}$ that ascertained the projection method and the SIFT-like algorithm are represented as P_{proj} and P_{sif} , respectively. Given this context, we can define a threshold value h_2 . Thus, the value of $P_{flaw-3d}$ is articulated as follows:

$$P_{flaw-3d} = \begin{cases} P_{proj}, & \text{if } \|P_{proj} - P_{sif}\| > h_2, \\ P_{sif}, & \text{if } \|P_{proj} - P_{sif}\| \leq h_2. \end{cases} \quad (6)$$

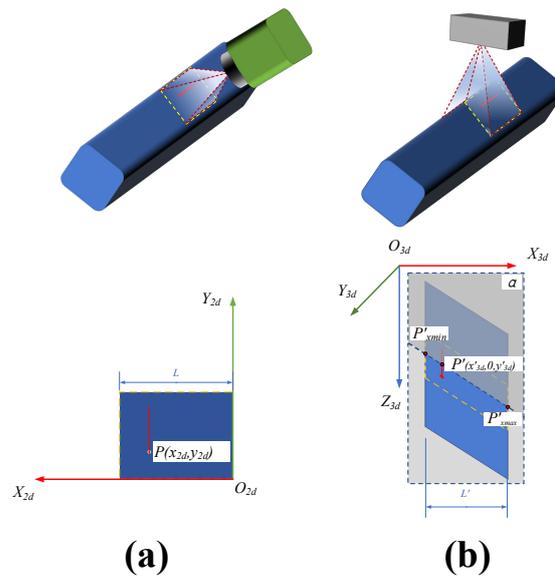


Figure 5. The method for a flaw’s position in one 2D camera converting to that in the 3D camera. In (a), the upper image represents taking Camera B as an example, and the lower image is a schematic of the corresponding image including the coordinate system; in (b), the upper image shows a schematic of the 3D camera’s capture, and the lower image is a point cloud schematic of Surface B including the coordinate system.

4.3. Marking

This section focuses on the computation of the coordinate transformation between the 3D camera and the manipulator. Specifically, the goal is to determine a transformation matrix that converts the coordinates provided by the 3D camera into the manipulator’s coordinate system. ICP is an algorithm widely used in computer vision and robotics to find the best correspondence and transformation between two sets of points. An ICP solve was introduced to address this challenge.

Acquisition of a point cloud image by the 3D camera ranges between 500 ms and 1000 ms, while the vision algorithms of the proposed system require approximately 800 ms. As a result, a specific installation gap between the 3D camera and the manipulator’s marking position is necessary. Our design incorporates a 1.8 m interval between them. Given the billet’s running speed of 0.2 m/s, this setup satisfies the system’s requirements.

The manipulator’s marking method involves holding a piece of chalk and pressing it onto the billet. The relative motion between the billet and chalk creates the marking line. Upon the encoder measuring the flaw’s length, the chalk is raised. The operational range of the marking manipulator with the chalk is depicted in Figure 3(d-1).

In actual marking trials to enhance response time, the marking system’s host computer, upon identifying the surface label, merely positions the manipulator to hover above the

respective surface's width center, awaiting instructions. After the coordinates are processed, the manipulator conducts minor movements to finalize the marking.

5. Algorithms and Their Realizations

To process digital operations, the raw point cloud data captured from the 3D camera need preliminary handling. As is observable in Figure 3(c-1), the initial point cloud image exhibits two surfaces, A and B, alongside several extraneous points which are potential sources of interference. The system's strategy is to bifurcate the point cloud into two segments using the region growing algorithm and subsequently filter out anomalies with the RANSAC algorithm.

5.1. Region Growing, Ransac, and Their Realizations

5.1.1. Region Growing Algorithm

For decades, region growing has been a fundamental technique in medical image segmentation [17]. It also finds its applications in the processing of 3D point cloud images [18,19].

The fundamental objective of the region growing technique is to aggregate point clouds with analogous properties to delineate a coherent region. For every distinct region intended for segmentation, an initial seed point is identified to kickstart the expansion. Adjacent points, showcasing similar attributes to the seed, are amalgamated into a singular cluster. This iterative accretion continues until no additional points fulfilling the preset criteria can be integrated, culminating in a well-defined region. The sequential steps are elucidated as Algorithm 1:

Algorithm 1 Region growing

Require: D : Data set, Model requirements, T : Specified iteration times

Ensure: M_{best} : The model with the most internal points

```

1: Initialize  $G_{\text{max}}$  to 0
2: Initialize  $M_{\text{best}}$  to null
3: for  $i = 1$  to  $T$  do
4:   Randomly select some data from  $D$  into set  $I$ 
5:   Build a mathematical model  $M$  using  $I$ 
6:   for each data point  $d$  in  $D \setminus I$  do
7:     Test the data point  $d$  with the model  $M$ 
8:     if  $d$  applies to  $M$  then
9:       Consider  $d$  as an internal point
10:    else
11:      Consider  $d$  as an external point
12:    end if
13:  end for
14:  Reestimate the model  $M$  using all hypothetical internal points
15:   $C_{\text{inliers}} \leftarrow$  Count of internal points for model  $M$ 
16:  if  $C_{\text{inliers}} > G_{\text{max}}$  then
17:     $G_{\text{max}} \leftarrow C_{\text{inliers}}$ 
18:     $M_{\text{best}} \leftarrow M$ 
19:  end if
20: end for return  $M_{\text{best}}$ 

```

5.1.2. RANSAC

The RANdom SAMple Consensus (RANSAC) algorithm is often used to solve plane detection tasks in computer vision, and it is also often used to remove external points in 3D point cloud images. The principle and process are as described in Algorithm 2:

Algorithm 2 RANSAC

Require: D : Data set, Model requirements, T : Specified iteration times

Ensure: M_{best} : The model with the most internal points

```

1: Initialize  $G_{max}$  to 0
2: Initialize  $M_{best}$  to null
3: for  $i = 1$  to  $T$  do
4:   Randomly select some data from  $D$  into set  $I$ 
5:   Build a mathematical model  $M$  using  $I$ 
6:   for each data point  $d$  in  $D \setminus I$  do
7:     Test the data point  $d$  with the model  $M$ 
8:     if  $d$  applies to  $M$  then
9:       Consider  $d$  as an internal point
10:    else
11:      Consider  $d$  as an external point
12:    end if
13:  end for
14:  Reestimate the model  $M$  using all hypothetical internal points
15:   $C_{inliers} \leftarrow$  Count of internal points for model  $M$ 
16:  if  $C_{inliers} > G_{max}$  then
17:     $G_{max} \leftarrow C_{inliers}$ 
18:     $M_{best} \leftarrow M$ 
19:  end if
20: end for return  $M_{best}$ 

```

5.1.3. Realizations

The Point Cloud Library (PCL) [20] can be used to easily call processing algorithms related to a point cloud.

Figure 6 displays how algorithms are realized. The program starts with obtaining a point cloud from the 3D camera, after which the point cloud could be cut into a smaller box. Subsequently, following calculating normals and curvature, the region growing method is used to merge points. Then, the RANSAC method is utilized to extract surface A or B. Moreover, the surface is projected into a horizontal one, which is the same as the plane captured by the flaw detection camera. Finally, the starting position of the flaw in the 3D camera’s coordinate system is calculated according to (1).

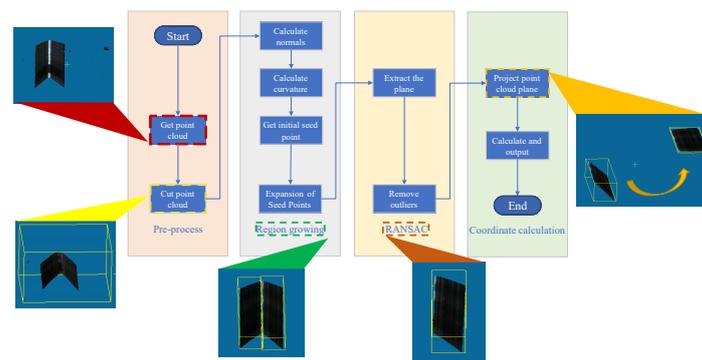


Figure 6. Realization workflow.

5.2. ICP Solve

We assume that there is a set of points whose coordinates are $P = \{p_1, p_2, \dots, p_n\}$ in the manipulator coordinate system. In the 3D camera coordinate system, these correspond to $P' = \{p'_1, p'_2, \dots, p'_n\}$. To seamlessly translate the coordinates deduced by the vision

program into the manipulator's coordinates, it is imperative to incorporate a Euclidean transformation into the program's configuration file, ensuring the following relationship:

$$\forall i, p = Rp'_i + t. \quad (7)$$

This problem can be solved by iterative closure point (ICP) [21]. The steps of the ICP solve are as described in Algorithm 3:

Algorithm 3 ICP solve

Require: Two groups of registered points

Ensure: Rotation matrix R and translation matrix t

- 1: Calculate centroid coordinates p and p' of the two groups
- 2: **for** each point p_i in the first group and p'_i in the second group **do**
- 3: Calculate eccentricity coordinates as

$$q_i = p_i - p, \quad q'_i = p'_i - p'. \quad (8)$$

4: **end for**

5: Solve for the rotation matrix R according to:

$$R^* = \arg \min_R \frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|^2. \quad (9)$$

6: Calculate the translation matrix t as

$$t^* = p - Rp'. \quad (10)$$

return R, t

In Step 5, SVD is applied to solve R [22,23]. It is a mathematical method used to break down a matrix to reveal its most important elements. The realization of the ICP solve is shown in the Experiments section.

6. Experiments

The equipment requirements and development environment utilized in the experiments are illustrated in Tables 1 and 2.

Table 1. Hardware conditions.

Item	Quantity	Model/Condition
Manipulator	1	HuaMing, HMQR5
2D camera	2	JAI, GO-2401M-PGE
3D camera	1	Chinshine, Surface HD50
Experimental site	1	A billet production workshop

Table 2. Software environment.

Item	Library/Software
Operating system	Ubuntu 18.04
UI development	QT 5.12
Point cloud processing	PCL 1.8 and Cloud Compare
Image processing	OpenCV 3.2

6.1. Calculation of Coordinate Conversion Matrix between 3D Camera and Manipulator

Experiment process: Make the manipulator mark 6 points on a billet (as is shown in Figure 7), and record the coordinates of these 6 points on the teaching pendant as

$\{p_1, p_2, \dots, p_6\}$. Then, use Cloud Compare software to obtain the coordinates of these 6 points in the generated point cloud (as is shown in Figure 8), which are recorded as $\{p'_1, p'_2, \dots, p'_6\}$. Table 3 describes the coordinates of the points obtained. Then, solve R and t according to Algorithm 3.



Figure 7. Points being marked by the manipulator on a billet. In total, there are 6 such points.



Figure 8. The 6 points taken by the 3D camera. Labeled from right to left as No. 1–6.

Table 3. p_k and p'_k .

k	p_k	p'_k
1	(0.440, −699.614, −609.065)	(51.866, 8.656, 551.423)
2	(−7.854, −707.887, −584.550)	(38.819, −7.414, 536.600)
3	(−22.004, −710.480, −565.058)	(20.969, 10.040, 523.670)
4	(−70.521, −681.394, −547.310)	(−27.320, −22.966, 527.000)
5	(−99.288, −679.490, −577.388)	(−45.017, −11.765, 546.945)
6	(−118.263, −703.891, −568.578)	(67.579, 0.378, 566.353)

Calculation results:

$$R = \begin{bmatrix} 0.9407 & -0.0332 & -0.3376 \\ 0.0195 & -0.9882 & 0.1516 \\ -0.3387 & -0.1492 & -0.9290 \end{bmatrix},$$

$$t = \begin{bmatrix} 134.3108 \\ -783.6481 \\ -74.1692 \end{bmatrix}.$$

Substitute R and t into the program to complete the automatic conversion of flaw points from the 3D camera coordinate system to the manipulator coordinate system.

6.2. Overall Dynamic Test

Experimental procedure: To visually represent the flaws, we simulated them by drawing lines on the billet using a black pen and measured their position in the flaw detection cameras. We then measured their positions inputting this data into the marking system, and subsequently initiated the billet's movement. After marking, we assessed

whether the chalk line adequately covered the black line and gauged the deviation between their center lines. Comparative experiments were conducted both with and without the 3D camera in the system. Initial calibration between the flaw detection cameras and the manipulator was performed at the commencement of these tests. The outcomes of the 10 experiments, employing the proposed method, are collected in Table 4. The results of the control experiments can be found in Figures 9 and 10. Figure 9 illustrates the deviations between the center lines of simulated flaws and those marked by the manipulator. Figure 10 exemplifies the methodology employed in measuring the deviations between these center lines, along with a subset of the results obtained.

Table 4. Experimental results.

Data Number	Surface	x_{2d}	Calculated Coordinates (x_m, y_m, z_m)
1	A	55	(−83.8226, −701.789, −567.817)
2	B	81	(−9.03072, −699.722, −598.169)
3	A	58	(−86.1592, −701.561, −568.805)
4	B	73	(−12.1695, −700.716, −590.882)
5	A	71	(−96.2829, −700.575, −573.082)
6	B	22	(10.9795, −693.383, −644.617)
7	A	34	(−67.4687, −703.382, −560.907)
8	B	46	(3.5686, −695.73, −627.413)
9	A	92	(−112.637, −698.982, −579.992)
10	B	45	(7.7073, −694.736, −611.699)

From the presented figures, it is evident that the system without the 3D camera fails to fully cover any of the flaws during the experiments. Conversely, the system equipped with the 3D camera demonstrates enhanced effectiveness.

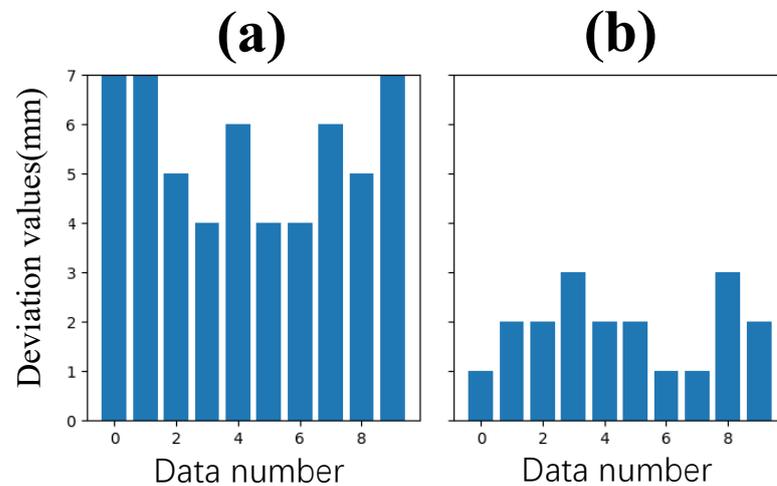


Figure 9. The chart depicts the deviation values of 10 trials for both the initial and improved experiments as the result. (a) Initial experiments. (b) Improved experiments. It suggests that our method yields more precise and stable results.

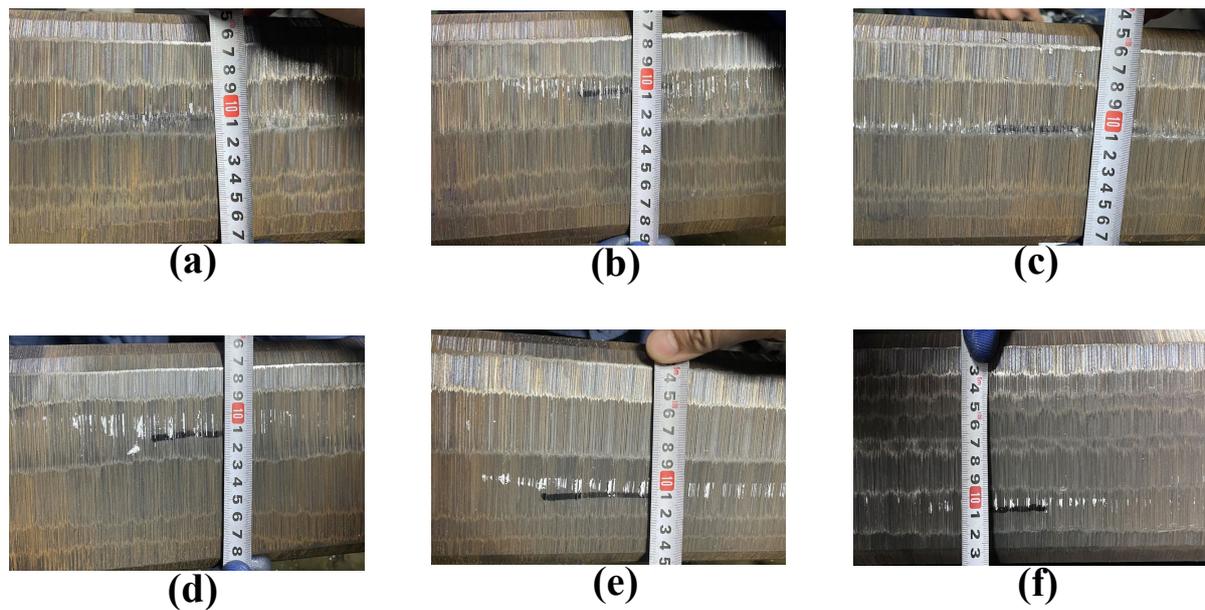


Figure 10. Comparison of results. (a–c)—Three results selected from the experiments conducted on the improved method (with our relocalization method). (d–f)—Three results selected from the experiments conducted on the initial method (without our relocalization method).

7. Conclusions and Future Work

Our study marks a significant advancement in the field of industrial flaw detection with the development of a novel automated system for detecting, relocalizing, and marking flaws on steel billets. The core innovation of our system lies in its utilization of point cloud data and a 3D camera, which collectively enable a more nuanced and precise identification of flaws compared to traditional image processing techniques. This technological leap not only enhances the accuracy of flaw detection, but also substantially reduces the reliance on labor-intensive processes and deep learning models that typically demand extensive computational resources and time for training.

Our experiments, conducted in a real-world steel production environment, demonstrated the system's efficacy and robustness. The ability to accurately track and mark flaws on moving billets, despite varying orientations and environmental conditions, underscores the system's operational viability in a fast-paced industrial setting. Furthermore, the system's precision in marking directly correlates to improved quality control measures, ensuring higher standards in the final steel products.

Looking ahead, integrating this system with automatic polishing mechanisms could pave the way for a fully automated production line, thereby extending its applicability. Additionally, while deploying deep learning necessitates substantial model training, its implementation can significantly enhance the system's functionality and robustness. For example, deep learning can be applied for advanced defect classification, enabling the system to distinguish between various types of surface flaws effectively.

Author Contributions: Conceptualization, H.Z. and Z.L.; methodology, H.Z.; software, H.Z. and Q.H.; validation, H.Z., J.C. and Z.L.; formal analysis, H.Z.; investigation, H.Z.; resources, Z.L.; data curation, X.Z.; writing—original draft preparation, H.Z.; writing—review and editing, Q.H., X.Z., J.C. and Z.L.; visualization, X.Z.; supervision, J.C. and Z.L.; project administration, Z.L.; funding acquisition, Z.L. All authors have read and agree to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wei, Y.; Zhang, Y.; Han, J.; Zhang, L.; Zhao, Z.; Shen, X. Surface Crack Detection Technologies for Steel Billet. *Mater. Rev.* **2016**, *30*, 75–79.
2. Zolfaghari, A.; Zolfaghari, A.; Kolahan, F. Reliability and sensitivity of magnetic particle nondestructive testing in detecting the surface cracks of welded components. *Nondestruct. Test. Eval.* **2018**, *33*, 290–300. [[CrossRef](#)]
3. Ghorai, S.; Mukherjee, A.; Gangadaran, M.; Dutta, P.K. Automatic defect detection on hot-rolled flat steel products. *IEEE Trans. Instrum. Meas.* **2012**, *62*, 612–621. [[CrossRef](#)]
4. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [[CrossRef](#)]
5. Luo, Q.; Sun, Y.; Li, P.; Simpson, O.; Tian, L.; He, Y. Generalized completed local binary patterns for time-efficient steel surface defect classification. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 667–679. [[CrossRef](#)]
6. Di, H.; Ke, X.; Peng, Z.; Dongdong, Z. Surface defect classification of steels with a new semi-supervised learning method. *Opt. Lasers Eng.* **2019**, *117*, 40–48. [[CrossRef](#)]
7. Zhang, S.; Zhang, Q.; Gu, J.; Su, L.; Li, K.; Pecht, M. Visual inspection of steel surface defects based on domain adaptation and adaptive convolutional neural network. *Mech. Syst. Signal Process.* **2021**, *153*, 107541. [[CrossRef](#)]
8. Wang, W.; Lu, K.; Wu, Z.; Long, H.; Zhang, J.; Chen, P.; Wang, B. Surface defects classification of hot rolled strip based on improved convolutional neural network. *ISIJ Int.* **2021**, *61*, 1579–1583. [[CrossRef](#)]
9. Barone, F.; Marrazzo, M.; Oton, C.J. Camera calibration with weighted direct linear transformation and anisotropic uncertainties of image control points. *Sensors* **2020**, *20*, 1175. [[CrossRef](#)] [[PubMed](#)]
10. Yun, J.P.; Choi, S.; Seo, B.; Kim, S.W. Real-time vision-based defect inspection for high-speed steel products. *Opt. Eng.* **2008**, *47*, 077204. [[CrossRef](#)]
11. Liu, H.W.; Lan, Y.Y.; Lee, H.W.; Liu, D.K. Steel surface in-line inspection using machine vision. In Proceedings of the First International Workshop on Pattern Recognition, Tokyo, Japan, 11–13 May 2016; Volume 10011, pp. 187–191.
12. Shi, T.; Kong, J.Y.; Wang, X.D.; Liu, Z.; Zheng, G. Improved Sobel algorithm for defect detection of rail surfaces with enhanced efficiency and accuracy. *J. Cent. South Univ.* **2016**, *23*, 2867–2875. [[CrossRef](#)]
13. Li, S.; Luo, C.C. Research on straightness detection of steel strip edge based on machine vision. In Proceedings of the Journal of Physics: Conference Series, IOP Publishing, Mosul, Iraq, 21–22 April 2021; Volume 1820, p. 012063.
14. Daniels, J.I.; Ha, L.K.; Ochotta, T.; Silva, C.T. Robust smooth feature extraction from point clouds. In Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007 (SMI'07), Minneapolis, MN, USA, 13–15 June 2007; pp. 123–136.
15. Li, D.; Shi, G.; Wu, Y.; Yang, Y.; Zhao, M. Multi-scale neighborhood feature extraction and aggregation for point cloud segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 2175–2191. [[CrossRef](#)]
16. Wang, H.; Zhang, J.; Tian, Y.; Chen, H.; Sun, H.; Liu, K. A simple guidance template-based defect detection method for strip steel surfaces. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2798–2809. [[CrossRef](#)]
17. Hojjatoleslami, S.A.; Kittler, J. Region growing: A new approach. *IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc.* **1998**, *7*, 1079. [[CrossRef](#)] [[PubMed](#)]
18. Poux, F.; Mattes, C.; Selman, Z.; Kobbelt, L. Automatic region-growing system for the segmentation of large point clouds. *Autom. Constr.* **2022**, *138*, 104250. [[CrossRef](#)]
19. Li, Q.; Song, D.; Yuan, C.; Nie, W. An image recognition method for the deformation area of open-pit rock slopes under variable rainfall. *Measurement* **2022**, *188*, 110544. [[CrossRef](#)]
20. Rusu, R.B.; Cousins, S. 3D is here: Point cloud library (pcl). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
21. Gao, X.; Zhang, T.; Liu, Y.; Yan, Q. *14 Lectures on Visual SLAM: From Theory to Practice*, 2nd ed.; Publishing House of Electronics Industry: Beijing, China, 2017.
22. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, 698–700. [[CrossRef](#)] [[PubMed](#)]
23. Pomerleau, F.; Colas, F.; Siegwart, R. A review of point cloud registration algorithms for mobile robotics. *Found. Trends® Robot.* **2015**, *4*, 1–104. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.