Vision-Guided Camera Pose Estimation for Robotic Rebar Tying

Shaopeng Xu[1], Huiguang Wang[1], Xiaoyi Lv[1], Lu Deng[1,2], Bo Jin[1,3], and Jingjing Guo[1,2,*]

[1]College of Civil Engineering, Hunan University, China
[2]Hunan Provincial Key Laboratory for Damage Diagnosis for Engineering Structures, Hunan University, China
[3]Hunan Huda Construction Supervision Co., Ltd, Changsha, China
[*]Corresponding Author
xsp@hnu.edu.cn, whg0917@hnu.edu.cn, lyuxiaoyi@hnu.edu.cn, denglu@hnu.edu.cn, jinbo@hnu.edu.cn, [*]guojingjing@hnu.edu.cn

Abstract –

Rebar tying is a critical yet time-consuming process in construction, often criticized by high labor intensity, repetitive motions, and low efficiency. These challenges have led to the development of rebar tying robots, which offer a promising solution to automate and enhance the process. However, existing rebar tying robots are unable to deal with varying orientations of working surface in practical environments, making it difficult for cameras to consistently maintain an ideal pose. To address these challenges, this paper proposes a vision-guided camera pose estimation method. This method includes three steps: (1) image preprosessing, (2) feature point detection and matching, and (3) transformation matrix calculation. Through these steps, the optimal camera pose can be estimated from images captured at random initial poses, allowing the camera to autonomously adjust to its ideal pose. Furthermore, an image evaluation method is introduced, forming a feedback loop with the pose estimation process to ensure high-quality image capture. The proposed method achieves a 99.5% success rate for pose estimation within three attempts, with an average computation time of 1.05 seconds. This approach helps improve the efficiency and accuracy of rebar tying operations, facilitating the automation of the rebar tying process for planar rebar cages.

Keywords –

Rebar tying, Camera pose estimation, Computer vision, Feature point matching, Rebar cage

1 Introduction

Rebar tying is a labor-intensive and repetitive process in rebar product manufacturing, often leading to physical strain [1] and reduced efficiency. To mitigate these challenges, rebar tying robots have been developed, such as Tybot [2], TOMOROBO [3], and RBBD-Bot2.0 [4], reflecting a trend toward automation in rebar tying.

Most rebar tying robots rely on vision guidance for their operations [5,6,7], mainly dealing with horizontal rebar meshes, where the camera is fixed to the robot body and does not require pose adjustments. However, in practical applications, rebar cages may be placed in various orientations, such as horizontally for composite slabs and vertically or inclined for T-beams and box girders in bridges, requiring adaptive adjustments to the camera pose. A 6-Degree-of-Freedom (DoF) robotic arm allows flexible positioning, ensuring that the camera maintains a perpendicular view of the rebar mesh, as shown in Figure 1. Additionally, efficient rebar tying requires that (1) the captured rebar mesh area remains rectangular (as shown in Figure 1); (2) rebar intersection points do not align with image edges, minimizing overlap and optimizing tying efficiency. Manual camera adjustments are time-consuming and impractical in fast-paced environments, necessitating an autonomous pose estimation method.

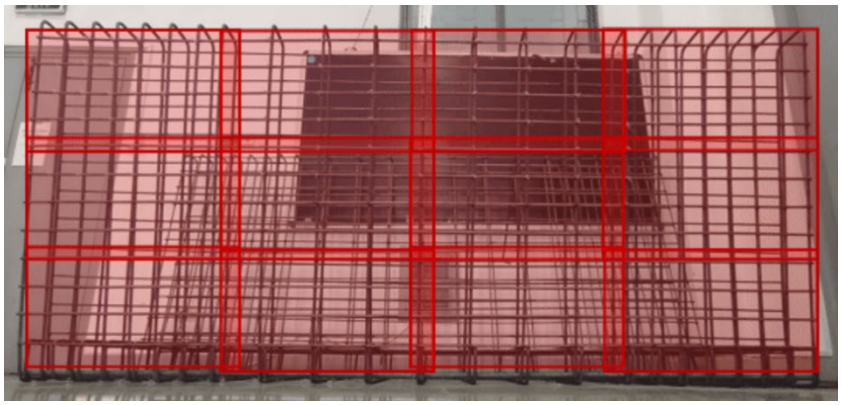

Figure 1. A rebar mesh divided into multiple rectangular work areas

Camera pose estimation involves determining the camera's position and orientation in 3D space, often using feature point matching techniques such as SIFT [8], SURF [9], and ORB [10]. However, these traditional methods struggle with rebar cages due to their uniform structure, leading to mismatches and inaccurate results.

Recent deep learning advancements, such as SuperPoint [11] and SuperGlue [12], have improved feature point detection and matching. SuperPoint provides high-quality keypoints and descriptors, while SuperGlue, based on Graph Neural Networks (GNN) [13], enhances matching accuracy. This study integrates SuperPoint, SuperGlue, and the Perspective-n-Point (PnP) algorithm to achieve accurate and reliable camera pose estimation.

To address the challenges posed by varying rebar cage orientations, this paper proposes a computer vision-based camera pose estimation method with two key innovations: (1) integrating a feature detector, a feature matcher, and PnP algorithm to determine the optimal camera pose from randomly initialized images, and (2) incorporating an image evaluation mechanism to establish a feedback loop for autonomous pose adjustment. This enables rebar tying robots to achieve optimal camera pose adjustment, improving operational efficiency and accuracy.

2 Method

The proposed method comprises three modules as illustrated in Figure 2: (1) image preprocessing; (2) the 6-DOF pose estimation of the camera; (3) image evaluation.

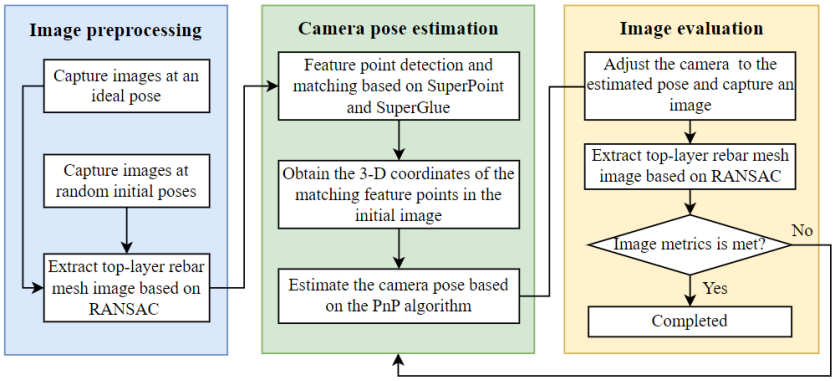

Figure 2. The workflow of the proposed method

2.1 Image Preprocessing

The proposed method requires two inputs: an RGB image and depth map, captured in an ideal pose and random initial pose, respectively. The RGB image and depth map captured in random initial poses are intended to consider the effects from the unanimous ground vehicle (UGV), which will be deviated due to its low positioning accuracy and complicated construction sites. This deviation causes the Z axis of the structured light camera, mounted at the end of the robotic arms, hard to maintain perpendicular to the top-layer rebar mesh of

the rebar cage.

The raw RGB image captured by the structured light camera contains excessive background information, making image matching and rebar crosspoints recognition difficult. To address this, it is essential to filter the background information of the raw RGB image. The process follows these steps: (1) align the captured RGB image and depth map to obtain the 3D point cloud of the rebar cage; (2) the point cloud of the top-layer rebar mesh is contained in the plane closest to the camera's optical center, which is the origin of the camera coordinate system; (3) a method based on RANdom SAmple Consensus [14] (RANSAC) is adopted to fit planes; (4) the plane closest to the camera's optical center is extracted.

After extracting the plane closest to the optical center of the camera, the point cloud belonging to this plane can be obtained. The pixel coordinates corresponding to the point cloud date can be calculated by depth values and the camera's intrinsic matrix, and an RGB image with background information removed can be obtained.

2.2 Camera Pose Estimation

This section describes a method for estimating the optimal camera pose based on RGB images and depth maps captured from a random initial pose. Initially, we perform feature point detection and matching on two filtered RGB images, which have excluded background information. Then, based on the coordinates of these matched feature points, the PnP algorithm is used to obtain the estimated camera pose.

2.2.1 Detect and Match the Feature Points

Recently, a significant number of algorithms have been substantially applied to detect the feature points between two images captured in different poses, such as FAST corner detector [15], SIFT, ORB, etc. However, these algorithms cannot cope with RGB images in complex scenarios, which exhibit poor robust during the process of detecting. Therefore, we adopt an approach based on neural network named SuperPoint, which can detect the feature points without manual annotations. After that, utilize a matching algorithm named SuperGlue to match feature points.

SuperPoint is a self-supervised network for training feature point and descriptors. In this framework, the authors trained a full convolutional network (FCN) based on synthetic dataset, which includes simple geometric shapes with no ambiguity in the feature point locations named MagicPoint. However, this pre-trained network performs poor performance in real dataset. As a result, they introduced Homographic Adaptation for boosting performance from synthetic dataset to real dataset. Finally, this framework can efficiently detect the location of feature points and provide descriptors for feature points. In this paper, we created a dataset consisting of 375 preprocessed rebar cage images as the training dataset for SuperPoint. These images were captured by the camera from different shooting distances and angles relative to the rebar cage. The image collection area is shown in Figure 3.

Figure 3. Image capture area of the SuperPoint training dataset

After detecting the positions of feature points and obtaining their descriptors, they are input into a GNN called SuperGlue. This network can match two sets of feature points using an attention-based flexible context aggregation mechanism.

2.2.2 Camera Pose Estimation

After extracting and matching feature points using the SuperPoint and SuperGlue algorithms, the pixel coordinates of the matched feature points in the initial image (filtered image obtained at a random initial pose) and the reference image (filtered image obtained at the ideal pose) can be obtained. By combining the depth map captured by the camera at the random initial pose, the 3D coordinates of the matched feature points in the initial image can be obtained. The input to the PnP algorithm includes the 3D coordinates of the matched feature points in the initial image and the pixel coordinates of the matched feature points in the reference image.

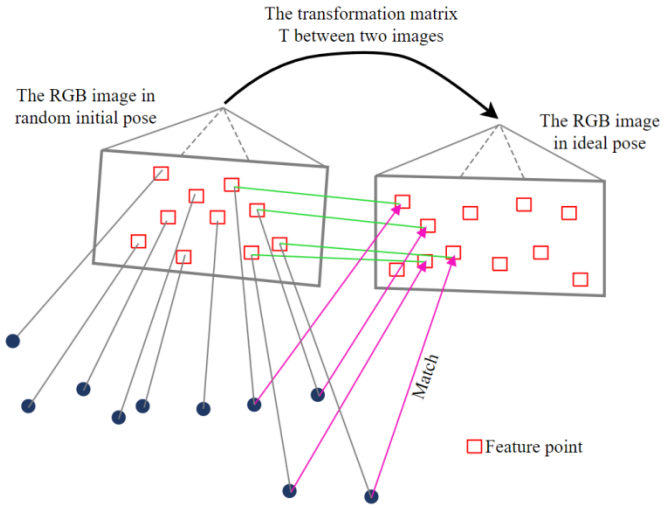

Figure 4. The core target of PnP algorithm

The core target of PnP algorithm is to get the transformation matrix from random initial pose to the ideal pose, which is illustrated in Figure 4. When getting the input of PnP algorithm, the transformation matrix T can be calculated based on Levenberg-Marquardt method [16]. As a result, the estimated pose can be determined by the equation below:

$$Pose_{estimated} = Pose_{initial} \cdot T \quad (1)$$

2.3 Image Evaluation

After obtaining the estimated pose of the camera through the PnP algorithm, move the camera to the estimated pose and collect images, and preprocess the original images to obtain the top-layer rebar mesh image. In order to ensure that the image meets the requirements, the image needs to be evaluated. The specific process is as follows: (1) the rebar crosspoints in the top-layer rebar mesh image are identified and located based on the YOLOv8-pose [17] keypoint detection algorithm; (2) find the maximum and minimum pixel coordinates along the u-axis and v-axis among all the rebar crosspoints, i.e., u_{max}, u_{min}, v_{max}, and v_{min}, as shown in Figure 5; (3) three image evaluation metrics can be calculated to assess the quality of filtered RGB images, where 'a' denotes the pixel width of the image and "b" represents the pixel height of the image.

$$delta\_u = abs(1 - \frac{u_{max}}{a} - \frac{u_{min}}{a}) \quad (2)$$

$$delta\_v = abs(1 - \frac{v_{max}}{b} - \frac{v_{min}}{b}) \quad (3)$$

$$sum\_u = 1 - \frac{u_{max}}{a} + \frac{u_{min}}{a} \quad (4)$$

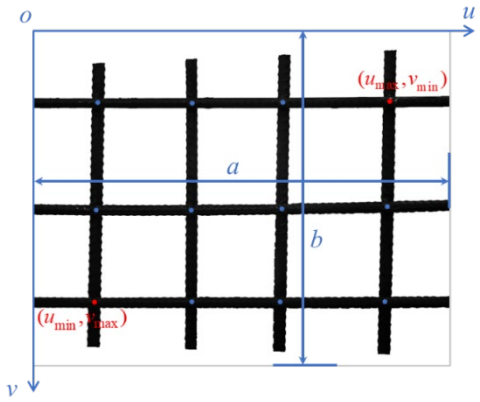

Figure 5. An ideal captured filtered RGB image

It is noticeable that all crosspoints are located on the central area of the filtered RGB image when $delta\_u < 0.08, delta\_v < 0.08$. Additionally, while all crosspoints are located on the center area of the RGB image, there are still a lot of places around the image that do not contain crosspoints. To make crosspoints are full of the RGB image, sum_u should be in a range from 0.1 to 0.4.

If the filtered image captured by the camera at the estimated pose meets the above metric requirements, the pose estimation is considered successful. Otherwise, if the captured filtered image does not meet the above requirements, it indicates that the camera has not yet reached its ideal pose. The filtered image captured by the camera at the current pose is used as input to the pose estimation module, and the camera pose estimation is repeated until the captured filtered image meets the above metric requirements.

3 Experiments and Results

3.1 Experiment Device

The proposed method was deployed on the rebar tying robot independently developed by our research group, as shown in Figure 6. This robot consists primarily of a mobile chassis, an industrial computer, and a 6-DOF robotic arm equipped with a structured light camera and a rebar tying actuator at the end. The structured light camera can simultaneously capture RGB images and depth maps, providing three-dimensional information of the captured objects. It is mounted at the end of the robotic arm in an "eye-in-hand" configuration, and the transformation matrix from the camera coordinate system to the robotic arm base coordinate system has been obtained through hand-eye calibration.

The industrial computer used in this study is configured as follows: Operating System: Ubuntu 20.04; CPU: 8-core ARM Cortex-A78; GPU: NVIDIA Ampere with 1,792 CUDA cores; RAM: 32GB. The development environment is primarily based on Python.

Figure 6. Rebar tying robot developed by our research group

3.2 Experiment Settings

The filtered rebar cage image shown in Figure 7 was obtained by gradually adjusting the camera pose, capturing images, and preprocessing the images. The evaluation metrics for this image is $delta\_u = 0.003, delta\_v = 0.026, sum\_u = 0.291,$ which meets the requirements of the image evaluation module. Thus, it is used as the reference image for the pose estimation module. Filtered images captured and preprocessed at all initial poses are matched with this reference image in the pose estimation module. When capturing this image, the optical center of the camera was 48 cm away from the plane of the top-layer rebar mesh. The z-axis of the camera coordinate system was perpendicular to this plane, the x-axis was parallel to the transverse rebar, and the y-axis was parallel to the longitudinal rebar.

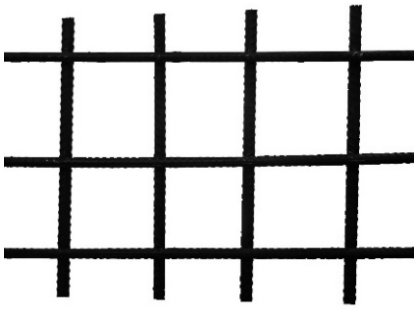

Figure 7. Reference image

To evaluate the performance of the pose estimation method when the camera is in different initial poses, the camera captured images from various shooting distances and angles as the input for the proposed pose estimation method. Additionally, to verify the robustness of the pose estimation method, the images used to test the method's performance were collected from different regions of the rebar cage, differing from the dataset used to train the SuperPoint network, as shown in Figure 3. And the design transverse rebar spacing and the design longitudinal rebar spacing of the rebar cage used in the experiment are both 10 cm.

After obtaining the initial image, the process for testing the performance of the pose estimation method is as follows: the initial image is input into the pose estimation module to calculate the estimated camera pose. The camera is adjusted to the estimated pose to capture an image, and the filtered image is input into the image evaluation module. If the image meets the required metrics, the camera pose estimation is complete, and the computation time for the pose estimation method is recorded. If the image does not meet the metrics, it is used as the initial image and input into the pose estimation module for re-estimation. This process is repeated until the captured image meets the required metrics, with the total computation time and the number of pose estimations recorded.

Considering that the recommended working distance of the structured-light camera is 30 cm to 60 cm, the camera's optical center is set at distances of 40 cm, 45 cm, and 50 cm from the plane of the top rebar mesh to ensure the depth values of rebar pixels in the filtered image fall within this range. This distance is hereafter referred to as the "shooting distance". Taking a shooting distance of 40 cm as an example, the camera pose is first adjusted so that the z-axis of the camera coordinate system is perpendicular to the plane of the top rebar mesh, the x-axis is parallel to the transverse rebar, and the y-axis is parallel to the longitudinal rebar. This pose is referred to as its "standard pose", as shown in Figure

8. Then, the camera can be rotated around its x-axis, y-axis, and z-axis respectively by the following angles: $-10°, -5°, 0°, 5°, 10°$ (counterclockwise is positive, clockwise is negative). Therefore, the camera has a total of 125 initial poses at this shooting distance, and a total of 375 initial poses at the three shooting distances.

Figure 8. Standard pose of the camera

3.3 Experiment Results

For each initial pose scenario, the following parameters are recorded: whether it can be adjusted to the ideal pose, the number of pose estimation attempts required to reach the ideal pose, and the computation time of the pose estimation method. The results are shown in Table 1. It can be seen from Table 1 that for the total of 375 initial pose scenarios at three shooting distances, the success rate of the pose estimation method within three attempts is 99.5%, and the success rate with just one attempt is 91.5%.

Table 1 Number of pose estimation attempts, corresponding scenario counts, and average computation time

Number of pose estimation attempts and the corresponding scenario counts							
Shooting distance (cm)	1	Success rate (%)	2	Success rate (%)	3	Success rate (%)	Failure
40	114	91.2	8	97.6	1	98.4	2
45	115	92.0	9	99.2	1	100.0	0
50	114	91.2	9	98.4	2	100.0	0
Total	343	91.5	26	98.4	4	99.5	2
Average computation time of pose estimation method (run once)						1.05s	

The failure of a single pose estimation attempt can be attributed to the large number of incorrect matching points in the initial image matching results, which leads to inaccurate pose estimation by the PnP algorithm. Consequently, the first estimated image fails to meet the evaluation metrics, necessitating additional pose estimation attempts. If the estimated pose renders the robotic arm unreachable or causes collisions, the pose estimation is considered a failure. The computation time in this study refers to the duration from receiving an image captured at an initial camera pose to generating the corresponding optimal camera pose that meets the predefined requirements. The proposed method achieves an average computation time of 1.05 seconds per pose estimation, meaning that the system can determine the desired camera pose within 1.05 seconds after receiving the initial image. This efficiency ensures minimal impact on production scheduling in precast plants and construction sites, making the method suitable for real-world deployment.

Figure 9 shows the test results of three initial pose scenarios, each of which includes: an image captured and preprocessed by the camera at the initial pose (referred to as "initial image"), the matching result with the reference image, and an image captured and preprocessed at the estimated pose (referred to as "estimated image"). In Figure 9(a), the initial pose of the camera for the initial image is: the shooting distance is 40 cm, and the camera rotates 10°, -10°, and -10° around its own x-axis, y-axis, and z-axis based on the standard pose. The match results show the matching between the initial image and the reference image. The image pose estimation module only requires a single run with a computation time of 1.062 seconds. The evaluation metric value for the estimated image is: $delta\_u = 0.005, delta\_v = 0.002, sum\_u = 0.270$, which meets the requirements. It can be concluded that even when the camera's initial pose significantly deviates from its ideal pose, the proposed pose estimation method can still calculate an ideal pose, enabling the camera to capture rebar cage images that meet the evaluation metrics.

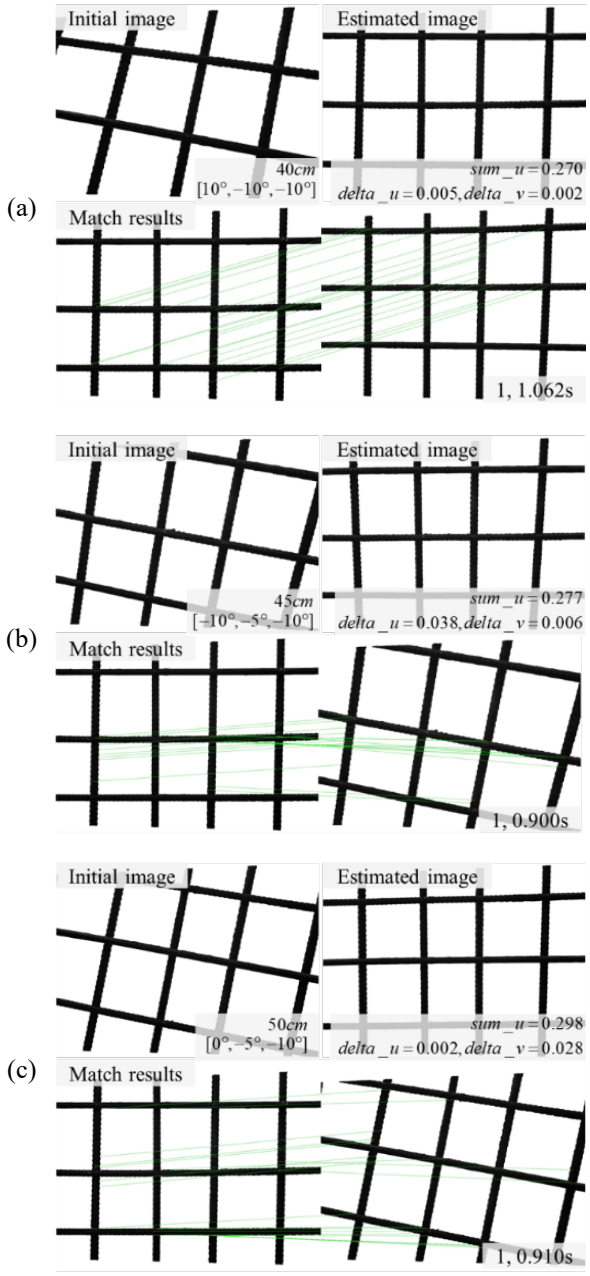

Figure 9. The initial image, matching result, and estimated image corresponding to the camera in different initial pose

4 Conclusion

This paper proposes a vision-guided camera pose estimation method for rebar tying robots, designed to calculate the ideal camera pose. This ideal pose facilitates efficient and high-quality rebar tying for rectangular planar rebar cages. The key innovations of the proposed method are as follows:

(1) A feature point-based camera pose estimation method is proposed. For the 375 initial pose scenarios tested in this study, the method achieves a 99.5% success rate for pose estimation within three attempts, with a 91.5% success rate for a single attempt. The average computation time for each estimation is 1.05 seconds, which is generally sufficient for practical applications.

(2) A method for evaluating the image quality captured by the rebar tying robot's camera is proposed. This method incorporates three evaluation metrics and forms a feedback loop with the camera pose estimation process, further ensuring the quality of the captured images. As a result, it contributes to the efficient and high-quality completion of rebar tying tasks for planar rebar cages.

However, the proposed pose estimation method requires multiple adjustments of the camera under certain initial pose scenarios and is only applicable to planar rebar cages. It is necessary to develop more efficient and robust pose estimation methods applicable to curved rebar cages and corresponding image evaluation metrics in the future.

Acknowledgment

The authors would like to acknowledge the financial support provided by the National Key Research and Development Program of China (Grant No. 2023YFC3806800), the National Natural Science Foundation of China (Grant No. 52308312), the Hunan Province Funding for Leading Scientific and Technological Innovation Talents (Grant No. 2021RC4025), and the Central Guidance for Local Science and Technology Development Fund Project (Grant No. 246Z5404G).

References

[1] M. F. Antwi-Afari, S. Anwer, W. Umer, H. Y. Mi, Y. Yu, S. Moon, and M. U. Hossain. Machine learning-based identification and classification of physical fatigue levels: A novel method based on a wearable insole device. International Journal of Industrial Ergonomics, 93:103404, 2023.

[2] Tybot. TyBot: REBAR TYING ROBOTS. On-line: https://www.constructionrobots.com/tybot, Accessed: 25/12/2024.

[3] TOMOROBO. REBAR TYING ROBOT TOMOROBO. On-line: https://en.kenrobo-

tech.com/tomorobo/, Accessed: 25/12/2024.

[4] RBBD-Bot2.0. The self-propelled intelligent rebar tying robot (RBBD-Bot 2.0) of China Construction Eighth Bureau made its debut. On-line: https://www.thepaper.cn/newsDetail_forward_20940885, Accessed: 25/12/2024.

[5] J. Jin, W. Zhang, F. Li, M. Li, Y. Shi, Z. Guo, and Q. Huang. Robotic binding of rebar based on active perception and planning. Automation in Construction, 132:103939, 2021.

[6] B. Cheng and L. Deng. Vision detection and path planning of mobile robots for rebar binding. Journal of Field Robotics, 41(6):1864–1886, 2024.

[7] R. Feng, Y. Jia, T. Wang, and H. Gan. Research on the System Design and Target Recognition Method of the Rebar-Tying Robot. Buildings, 14(3):838, 2024.

[8] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, 2:1150-1157, 1999.

[9] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision*, Part I 9: 404-417, 2006.

[10] V. E. Rabaud. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, 2564-2571, 2011.

[11] D. Detone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 224-236, 2018.

[12] P.-E. Sarlin, D. Detone, T. Malisiewicz, A. Rabinovich, and E. Zurich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4938-4947, 2020.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. IEEE Transactions on Neural Networks, 20(1): 61-80, 2009.

[14] M. A. Fischler and R. C. Bolles. Random sample consensus. Communications of the ACM, 24(6): 381-395, 1981.

[15] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision*, Part I 9: 430-443, 2006.

[16] H. P. Gavin. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering, Duke University, 2019.

[17] S. Wang, L. Deng, J. Guo, M. Liu, and R. Cao. Automatic quality inspection of rebar spacing using vision-based deep learning with RGBD camera. In *Proceedings of the International Symposium on Automation and Robotics in Construction*, 41:57-64, 2024.