Robotic Wiring Harness Bin Picking Solution Using a Deep-Learning-Based Spline Prediction and a Multi-stereo Camera Setup

Manuel Zürn[1](✉), Carsten Schmerbeck[2], Andreas Kernbach[3], Mara I. Kläb[3], Alper Yaman[3], Daniel Bragmann[4], Michael Heizmann[2], Marco Huber[3], Werner Kraus[4], Armin Lechler[1], and Alexander Verl[1]

[1] ISW, University of Stuttgart, Seidenstr. 36, Stuttgart 70174, Germany
manuel.zuern@isw.uni-stuttgart.de
[2] IIIT, Karlsruhe Institute of Technology, Hertzstraße 16, Karlsruhe 76187, Germany
[3] IFF, University of Stuttgart, Allmandring 35, Stuttgart 70569, Germany
[4] IPA, Fraunhofer Institute for Manufacturing Engineering and Automation,
Nobelstr. 12, Stuttgart 70569, Germany

Abstract. The automation of wire harness handling and installation in the automotive industry presents a challenge due to the inherent flexibility of cables, the high variance in wire harnesses and plug combinations, and the intricate spatial configurations required for accurate installation. Addressing this challenge requires the integration of sensors for accurate pose estimation with high-dexterity robotic systems. This work introduces a novel approach to automate the process of grasping of wiring harnesses for autonomous installation using a robotic arm. The methodology encompasses several stages. Initially, a multi-stereo camera setup creates a high-accuracy representation of the working area. Next, a deep learning model predicts a spline representing the segment with the biggest connector attached to it for 6D grasp pose estimation. The final stage uses a skill-based robot program to perform the grasping of the wiring harness, which is evaluated using 50 random configurations inside a bin. As a result, the proposed solution achieves an accuracy of 82% of successful wiring harness bin picking grasps, where success is defined when the result is that a specific connector on the wiring harness is in a predefined spot after grasping. Future work will use another robot to grasp the connector from the first robot to install it in an automotive demo door using reinforcement learning.

Keywords: Wire harness · Correspondence matching · Skill-based robotic control · Machine Vision · Deep Learning

M. Zürn and C. Schmerbeck—contributed equally to this work.

D. Holder et al. (Eds.): SCAP 2024, ARENA 2036, pp. 319–334, 2025.
https://doi.org/10.1007/978-3-031-88831-1_25

1 Introduction

The automotive wiring harness market is poised for continuous growth up to 2033, as indicated by market research [1]. This growth is driven by various factors, e.g., electrification and smart assistant systems. Considering the wiring harness value chain, costs are significant and encompass material, assembly, and installation. A bottleneck in the value chain is the final installation process, which consumes 70% of total production time and relies heavily on manual labor, with over 90% of tasks performed manually [2]. The manual-intensive process requires over-specification due to uncontrolled manipulation forces, which also may pose safety risks. Automation solutions emerge to address these challenges. By implementing automation, the industry aims to achieve several objectives: (i) Reduction of over-specification: Automation technologies can decrease dimensions and monitor manipulation forces, leading to a decrease in over-specification and minimizing material wastage. (ii) Decrease in manual installation time: Even partial automation solutions, e.g., human-robot collaboration, in wiring harness manipulation reduce stress on the worker due to the heavy automotive wiring harness. (iii) Enhanced Safety: Automation enables advanced monitoring systems that ensure precise and controlled installation, thereby improving safety standards and reducing the risk of errors or accidents.

Despite the benefits of automation, challenges persist in developing automation solutions, particularly in assembly tasks. Research from [3] underscores the complexities involved in wiring harness perception and dynamic robot programming, hindering the adoption of automation in this domain. Wiring harness manipulation requires different domain experts in 3D vision, advanced correspondence estimation for (i) wiring harness topology estimation and (ii) 6D component pose estimation and lastly advanced sensitive robotic control. This paper addresses a wiring harness bin-picking solution for manipulating medium-sized wiring harnesses by combining advanced multi-stereo 3D vision and wiring harness correspondence estimation with skill-based robotic control. The wiring harness is the size of a door wiring harness. We contribute a modular low-cost stackable camera system with high resolution for accurate 3D reconstruction, applying a recent implementation of generic intrinsic camera calibration [4], applying a modified iterative-closest-point algorithm [5] for extrinsic calibration in a multi-stereo approach, novel deep-learning architectures for estimating the topology of wiring harnesses predicting cubic splines and an evaluation of grasping a wiring harness in almost random positions in a bin. Furthermore, we contribute the dataset, model weights and the tensorboard log including quantitative and qualitative results in Supplementary information.

1.1 Related Works

Research in wiring harness perception mostly focuses on using 2D or 3D camera systems for perception with different goals. Applications range from visual inspection [6–8] to grasp point estimation on multiple wiring harnesses in a bin

[9] to robotized assembly [10,11] of wiring harnesses or artificial dataset augmentation [7,12]. Estimation of wiring harness states, e.g. done in [9,10], are done using probabilistic model estimation. Deep learning solutions have been applied for estimation segmentation masks [6,7] or directly valid grasp points in a bin as in [9]. However, solutions for segmentation masks usually have no information about the topology of the wiring harness required for robotic installation as in [10], as well as direct grasp point estimation as in [9].

There exists analytical topology estimation as in [13], which is limited to a restricted initial configuration of the wiring harness and therefore not directly suitable for bin picking. Solutions, as proposed in [11], which attach clamps with visual markers on the wiring harness, seem unpractical, as they introduce extra weight to the wiring harness. As concluded in [14], there exist vision-based solutions with deep learning offering a promising tool to solve these, it remains to be shown how accurate and how reliable the vision-based solutions perform. This study proposes and evaluates a deep learning approach for partially predicting the topology by predicting a specific segment of a wiring harness inside a bin, providing a reliable solution that tries to impose minimal constraints on the initial configuration of the wiring harness.

2 Methodology

2.1 Process Overview

The process setup is shown in Fig. 1. It consists of starting with a wiring harness in a random configuration in a bin. The bin is recorded from a multi-camera setup in different orientations and positions in a robotic cell. A deep learning model predicts a spline using 4 interpolated keypoints on each image. In the end, a grasp point on the wiring harness used for robotic wiring harness bin picking is estimated.

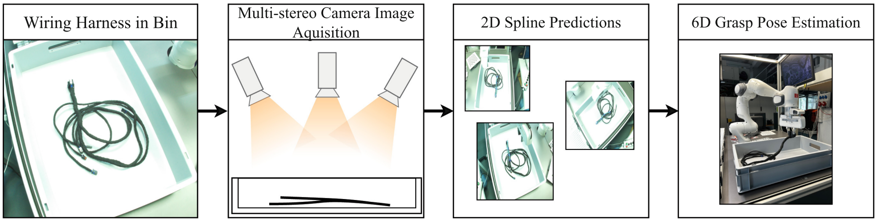
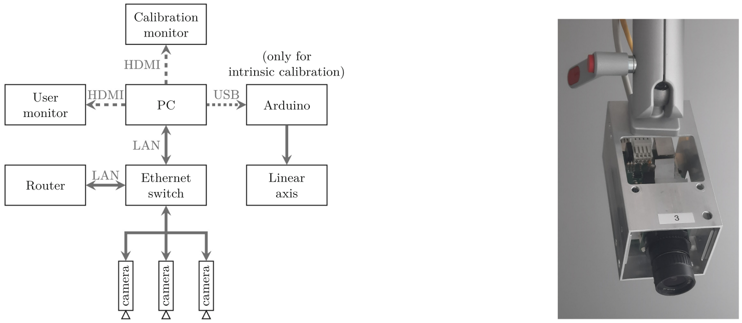


Fig. 1. Proposed wiring harness bin picking process. It consists of first, estimating three 2D splines representing a wiring harness segment and second estimating a 6D grasping pose.

2.2 Multi-stereo Camera System

A multi-stereo camera system is used to achieve accurate pose estimation of the wire harness. A passive vision system with multiple viewpoints was chosen because it does not require special illumination (such as structured light or laser scanners) [15] and the cameras can be directed or placed near the regions of interest in the scene, see Fig. 2. Furthermore, there is the possibility that the working area of the vision system is obstructed by, for example, the robot arm.



(a) Structure of the vision system

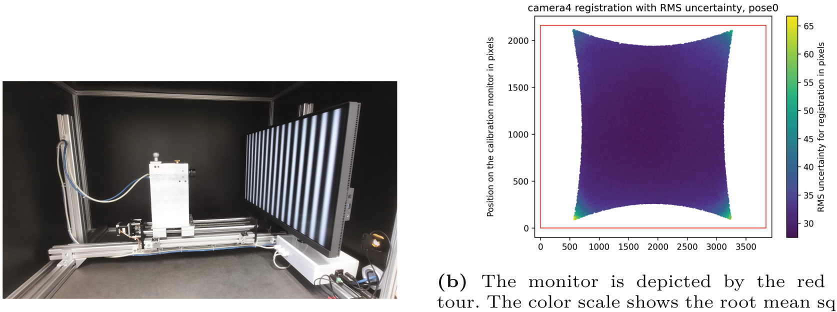
(b) Camera module with swivel mount

Fig. 2. Software and hardware setup of the camera modules

To achieve this, a modular camera system that allows the usage of a easily expandable number of cameras has been created, as illustrated in Fig. 2a. The mounting system enables the cameras to be securely mounted within a frame surrounding the robot cell, while still allowing for easy placement around the cell's perimeter. Each camera as shown in Fig. 2b utilizes a Raspberry Pi HQ camera with a C/CS lens mount, offering a resolution of 4056 × 3040 pixels and a 6 mm lens [16]. Additionally, the square design of the camera case permits stacking in any direction, facilitating stereo or light-field configurations. The experimental setup includes three camera modules aimed at the pick-up area where the cable is retrieved from a bin.

Generic Intrinsic Camera Calibration. The precise geometric calibration of camera systems is important for computer vision applications. Typically, these systems are modeled by using a perspective projection with a single projection center and are often characterized by low-dimensional, parametric models with minimal intrinsic parameters, such as the pinhole model [17, Chapter 2.3.1][4, Chapter 5.1]. The drawback of low-dimensional models lies in their limited descriptive capacity, which is insufficient to perfectly characterize every pixel among the vast number present in modern cameras. As a result, novel camera models have recently been developed. These generic models can be characterized as versatile imaging systems, independent of the used camera type, thereby enabling precise calibration [4].

The generic camera model was first introduced by Grossberg and Nayar [18]. In this model, any imaging setup is represented as a non-parametric discrete entity, like a black box, containing photosensitive components. Each individual



(a) Intrinsic calibration setup with sinusoidal pattern displayed on the monitor.

(b) The monitor is depicted by the red contour. The color scale shows the root mean square (RMS) error for the registration in monitor pixels.

Fig. 3. Calibration setup and example registration between camera and monitor.

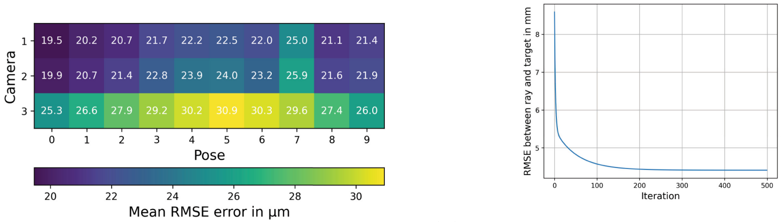
pixel within this system captures a ray of light, referred to as raxel, comprising geometric ray coordinates alongside radiometric parameters. The entirety of these raxels represents the imaging model of the camera. In this work, intrinsic camera calibration is performed in two steps for each camera depicted in Fig. 3a.

(i) Initially, sequential images of sinusoidal patterns with phase-shift coding shown on a monitor are captured (Fig. 3a), and the registration between the camera pixels and monitor coordinates is determined using probabilistic phase unwrapping (Fig. 3b). A modified version of Uhlig's implementation [4, Chapter 4] is used for this purpose. This approach is relatively robust against external factors such as lighting. However, its robustness is limited in cases of strong reflections or glare on the calibration monitor. Therefore, the calibration setup is enclosed in a box to block unwanted light from the outside. The recording of the monitor is done from 10 equally spaced distances to allow for the accurate estimation of the raxels. For ease of use, this process is automated by placing the camera in the calibration setup on a linear axis, as seen in Fig. 3a. The display of the 96 calibration images (48 per direction, 4 frequencies [1,4,16,64] and 12 phase angles), the recording, and the movement of the camera (10 poses) are automatically controlled [19]. The number of poses and configuration of images is identical to the one use in [4].

(ii) Thereafter, the intrinsic calibration is found by an alternating minimization-based process of both estimating the set of rays $\mathcal{L}$ intersecting the registered points on the different monitor poses and estimating the set of monitor poses (translation $\mathcal{T}$ and rotation $\mathcal{R}$). This can be formulated as a least-squares minimization problem:

$$f(\mathcal{R},\mathcal{T},\mathcal{L}) = \sum_{k,i} \mathrm{d}\left(\mathbf{p}_{ik},\mathbf{l}_i\right)^2 . \tag{1}$$

Here, index i represents the individual rays and index k the reference coordinate system. $d(\cdot)$ is a ray-to-point distance measure between points on the calibration monitor $\mathbf{p}_{ik}$ and $\mathbf{l}_i$ being a individual ray. $\mathbf{R}_k \in \mathrm{SO}(3), \mathbf{t}_k \in \mathbb{R}^3$ are the corresponding transformations to the camera coordinate system [4, Chapter 5].



(a) Intrinsic camera calibration, averaged over all raxels for each camera and pose.

(b) Example of RMSE distances for extrinsic calibration between calibration rays and calibration targets over 500 iterations in mm.

Fig. 4. Quantitative results of the intrinsic and extrinsic camera calibration

The intrinsic calibration process described by Uhlig [4] returns an RMSE for each ray at each monitor pose representing the distances between the registration on the monitor and the intersecting ray. To display them in Fig. 4a the individual rays were averaged per pose and per camera. The results between roughly 20 µm and 30 µm are significantly smaller than the pixel pitch of the monitor used for calibration of 185 µm (BenQ PD2705U) proving the subpixel accuracy capabilities stated by Uhlig [4]. In his work, an accuracy of 32 µm was achieved with different cameras and monitors [4].

Generic Extrinsic Camera Calibration. After the intrinsic calibration of each camera, an extrinsic calibration is performed. This is done by attaching a calibration target with ArUco markers to the robot end effector. The markers are found by an ArUco detector implementation in OpenCV [20]. The robot moves the calibration target to known positions while the camera system takes the corresponding pictures, resulting in known calibration points P in the 3D space. To compute extrinsic calibration, an initial guess for each camera pose relative to the known calibration points $\mathcal{P}$ is calculated using the SolvePnP method implemented in OpenCV [20]. This method employs a standard pinhole camera model where the camera parameters are derived exclusively from the camera and lens data sheet [16] and assume an ideal camera. This yields an initial solution for the subsequent iterative process.

The exact pose of the cameras is estimated using a modified point-to-line iterative closest point (ICP) algorithm. The generalized approach of an ICP algorithm was described by Besl&McKay [5]. The set $\mathcal{P}$ describes a set of known calibration points in the scene. Here they are the corners of the ArUco markers on calibration targets. Only the rays intersecting the known calibration points $\mathcal{P}$ are selected from the intrinsic calibration model and called calibration rays $\mathcal{X}$. Initial values are denoted with a subscript 0.

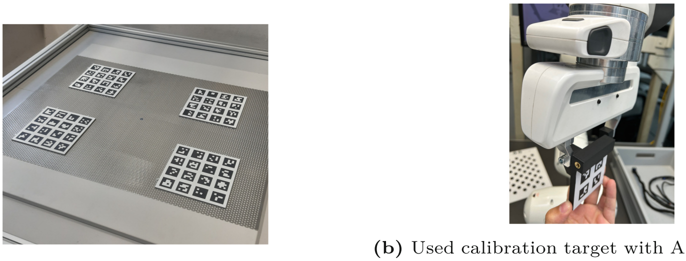
ICP is an iterative process, consisting of four steps:

1. **Compute the closest points** in the model shape $\mathcal{X}_k$ to their corresponding data points in $\mathcal{P}$. $\mathcal{Y}_k = c(\mathcal{P}, \mathcal{X}_k)$ denotes the set of closest points. Besl&McKay

describe their algorithm for a limited set of geometries [5], whereas in our implementation, the closest point on a ray is calculated [21, Chapter 10.2.1].

2. **Compute the registration**. The distances d_k between the closest points are calculated $(\mathbf{Q}_k, d_k) = \mathcal{Q}(\mathcal{P}_0, \mathcal{Y}_k)$ where $\mathbf{Q}_k$ denotes the registration. A implementation of Murugan [22] is used here.
3. **Apply the registration**. Besl&McKay apply the registration to the points that should be fitted to the model. $\mathcal{P}_{k+1} = \mathbf{Q}_k(\mathcal{P}_0)$ In our case, the model should be translated and rotated accordingly, to fit to the known points. $\mathcal{X}_{k+1} = \mathbf{q}_k(\mathcal{X}_0)$
4. **Repeat until termination**. Here, the termination condition is either a maximum number of steps performed or an RMSE below a threshold τ.

As illustrated in Fig. 4b, the proposed generic ICP method can substantially lower the RMSE distances compared to the initial solution provided by SolvePnP [20]. Preliminary results from a static scene with manually positioned and measured calibration targets (refer to Fig. 5a) indicate that this technique can iteratively decrease the RMSE reprojection error for 468 visible calibration points $\mathcal{P}$ to as low as 2.85 mm.



(a) Example of a static scene with ArUco markers used for extrinsic calibration.

(b) Used calibration target with ArUco markers. The target can be screwed into the robotic gripper to have a fixed transformation ${}^{EE}\mathbf{T}_{Target}$

Fig. 5. Markers for extrinsic calibration

Table 1. RMSE and maximum distances between the calibration points $\mathcal{P}$ and the calibration rays $\mathcal{X}$ after the extrinsic calibration.

Camera	RMSE in mm	Max in mm
1	7.80	17.65
2	6.54	12.25
3	4.41	10.59

As seen in Table 1, the results for the extrinsic calibration using the robot (see Fig. 5b) are slightly worse than using a static scene. This discrepancy could be caused by inaccuracies in the robot kinematics and requires further research to enable more accurate extrinsic camera calibrations. Nevertheless, the RMSE errors during the keypoint estimation are significantly lower than the values achieved during the extrinsic calibration (see Subsect. 3.1). Therefore one could assume that the inaccuracies in the robot kinematics may be mean-free and average themselves out by using multiple poses of the robot during the calibration.

2.3 2D Spline Prediction

Deep Learning Architecture. A deep learning model is trained to predict a spline on the first segment of the wiring harness. As the deep learning model predicts a specific segment on the wire harness, the process is referred to as partial direct topology prediction. The deep learning model consists of a backbone that extracts features of an RGB input image, a neck that extracts features from different feature map resolutions of the backbone and a head that predicts the interpolating cubic spline points of the spline. A reference image of the architecture is shown in Fig. 6. We used a tiny Swin Transformer from [23] as backbone. The backbone extracts features in 4 depth layers. By using a feature pyramid network [24] as neck, we ensure the capture of multi-scale feature representations which should improve model performance. Next, the different feature maps are concatenated and upsampled according to a U-Net architecture [25], which ensures faster learnability with skip connections.

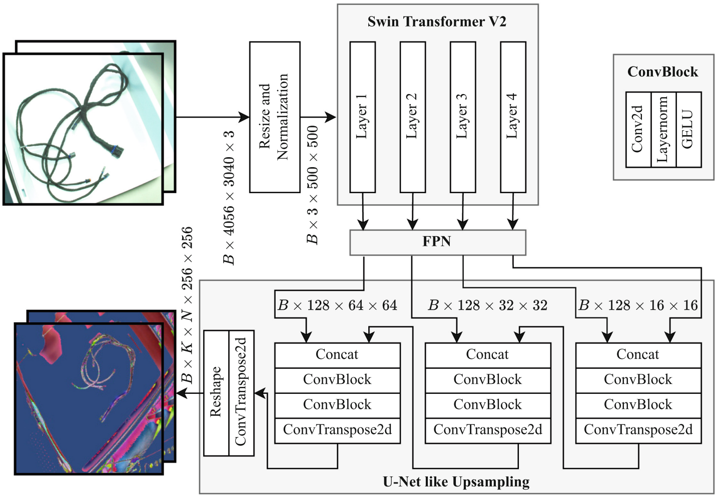


Fig. 6. Deep learning architecture using a Swin Transformer backbone, Feature Pyramid Network (FPN) Neck with a U-Net-like Convolutional Upsampling Head to predict natural cubic spline segments for wiring harnesses. This can be used to extract a partial directed graph for direct topology prediction as proposed in [13].

The output of the model is a heatmap with $B \times C \times H \times W$ features, where B corresponds to batch size, C to keypoint dimension times number of segments, H

to the height of the heatmap and W to the width of the heatmap. By reshaping C, we can learn different segments, such that the final dimension is $B \times K \times N \times H \times W$, where K corresponds to the number of segments and N to the number of keypoints on each segment. In this paper, we chose to learn one segment (K=1) modeled with 4 keypoints (N=4) to represent the cubic spline.

Loss Function and Training. Annotation takes place by approximating the shape of the segment using a natural cubic spline. The annotation tool was written in Python using NiceGui[1] and stores the segments and nodes in .json format. Standard deep learning praxis in e.g., facial landmark detection, use heatmaps to estimate keypoints or directly predict the keypoints by using regression and L1/L2 loss. Heatmaps tend to converge faster, while they usually can't be used for direct coordinate regression due to loss of gradient in non-differentiable keypoint extraction. Therefore, we use a way of estimating the keypoints without losing the gradient proposed in [26]. We apply L1 loss on the keypoint estimation and conducted a hyperparameter study with various variations of the loss functions and the hyperparameters. The hyperparameter study was conducted using

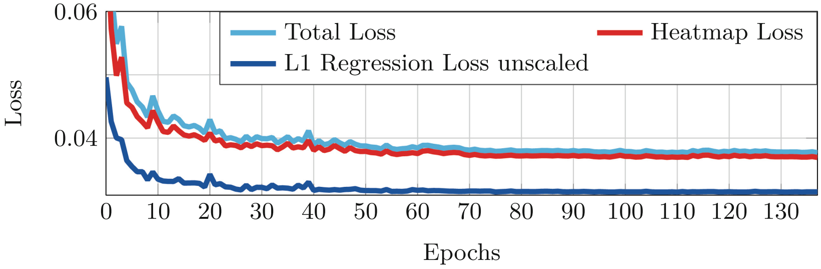


Fig. 7. Validation loss over a dataset with 584 annotated images. Hyperparameters: Batch size 5, learning rate 1.1e-4 with 139 epochs. Regression loss is scaled with 0.5 and added to the heatmap loss to get the total loss, which achieves the best results. Training/validation data split is 0.8/0.2.

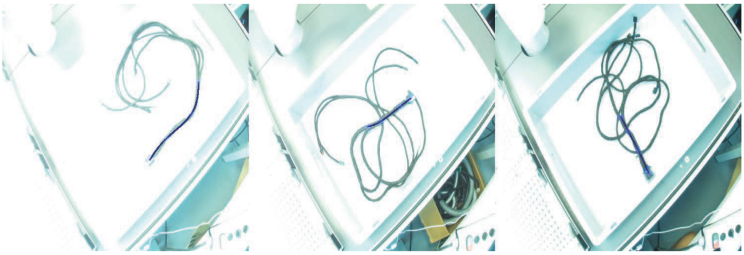


Fig. 8. Validation dataset samples of the natural cubic spline prediction (blue) depicted as overlay in the camera images. (Color figure online)

[1] doi.org/10.5281/zenodo.13623828.

optuna [27] with a hyperband pruner and a TPESampler for sampling in the learning rate between 10^{-5} and 10^{-3}, batch size between 5 and 20 and epochs between 100 and 300. The best trial is shown in Fig. 7. Samples of the prediction of the validation dataset are shown in Fig. 8.

2.4 3D Point Reconstruction and 6D Grasp Pose Estimation

For 3D point reconstruction we need the correspondence in each image represented by keypoints. The keypoints are given in pixel coordinates (s, t) as seen by the camera. For each keypoint, the corresponding ray from each camera is selected and stored together as a set of rays that should theoretically intersect at the sought-after keypoint. Finding a solution for the closest point to n (generally) skewed lines is done by a Nelder-Mead minimization of the sum of squared distances between the point and the set of lines [28]. Each 3D point reconstruction thus returns an RMSE value representing the uncertainty of the estimate. For robotic grasping, we need a 6D grasp pose consisting of an orientation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. Two 3D points and the $\mathbf{xy}$ plane are used to estimate the 6D pose. First, the connector point of the spline is used, as it is usually the one with the highest estimation certainty. Second, the first keypoint of the spline is used apart from the connector. Both are transformed into two 3D points using the intersection point of the rays as previously described. The orientation is restricted to an orientation about the $\mathbf{z}$ axis of the robot base, as the box $\mathbf{xy}$ plane is parallel to the one of the robot base coordinate system. $\mathbf{R}$ can then be calculated by subtracting one point from the other and projecting and normalizing the vector, resulting in the $\mathbf{x}$ axis of $\mathbf{R}$. As $\mathbf{z}$ is fixed, $\mathbf{y}$ can be calculated using the cross product of $\mathbf{z}$ and $\mathbf{y}$. The translational part is calculated by going $\frac{2}{3}$ of the distance between the estimated connector point and the estimated first keypoint in the keypoint direction, which is just an arbitrary position on the segment and has no particular meaning.

2.5 Skill-Based Robotic Architecture

The proposed wiring harness bin picking is evaluated by executing the states according to Fig. 9. Starting from a home position, a valid grasp point is estimated. The robot moves then and tries to grasp the valid position.

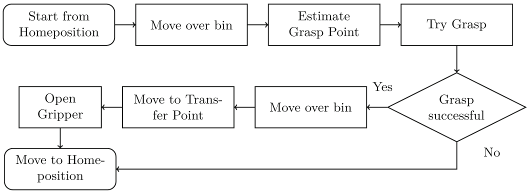


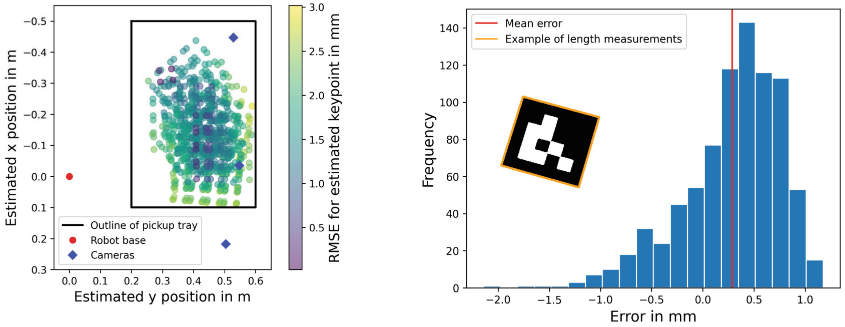
Fig. 9. Flow chart of the robot validation program. The different states in the flow chart are composed of parameterizable robot skills. The transfer point indicates a station where in future work a second robot grasps the connector of the wiring harness for plugging purposes.

Failure might occur due to the gripper colliding with the box, the robot moving into singularities due to limited rotation around its hand joint or due to collision of the gripper with the wiring harness resulting from incorrect grasp point estimations. The results are shown in Sect. 3.

3 Experiments

3.1 Validation of Keypoint Measurement Accuracy

When determining the positions of keypoints in space, it is of interest to achieve this goal as accurately as possible. Therefore, a scene with calibration targets randomly placed in the pickup box in 13 poses was prepared and recorded, see Fig. 10. The recordings were then evaluated for two different metrics: (a) The accuracy with which the three rays of the cameras intersect at the reference points and (b) the accuracy of length measurements in the scene. For both, the corners of the ArUco markers were used as reference targets.



(a) Scatterplot of estimated keypoints on a calibration target with 64 reference points in 14 poses, viewed from above. The color denotes the RMSE distance between the rays for the estimate and the resulting point.

(b) Histogram of the errors in mm while measuring distances of 40 mm on the calibration target with 64 reference points in 14 poses and an example of four length measurements.

Fig. 10. Validation of the 3D point estimation

(a) After estimating the position of the reference target, the distances between the rays and the corresponding estimated target positions were calculated and displayed as shown in Fig. 10a. (b) For this evaluation, the distances between the corners of the ArUco markers were calculated along their four sides, as depicted in Fig. 10. Then the actual length of the marker sides of 40 mm was subtracted from the measurements, resulting in an absolute error. One can see, that the average length measurement is longer than the actual value. This could be caused by the cameras being slightly out of focus, possibly resulting in OpenCV estimating the targets to be larger than they actually are. These results show that estimating keypoints and lengths in the scene is possible, but further research in this area is needed to investigate possible sources for errors and improve the quality of the measurements.

3.2 Wiring Harness Bin Picking

The experiment validates if the proposed uncertainties in the form of intrinsic and extrinsic calibration uncertainties and the uncertainties in the 2D spline prediction with 3D grasp point evaluation are low enough for autonomous robotic wiring harness bin picking applications. 50 consecutive trials according to the flowchart in Fig. 9 were recorded and evaluated using a Franka Emika Panda 7 Axis Cobot. Each time, the wiring harness is almost randomly positioned inside the bin. Three simplifications were used for placing the wiring harness inside the bin. (i) The pickable segment can't face in the direction of the robot base to avoid robotic kinematic hand singularities. (ii) The pickable segment can't be too close to the box's border to avoid collisions between the robotic hand and the box and (iii) the pickable segment is always on top of the wiring harness. Figure 11 shows 6 example trials.

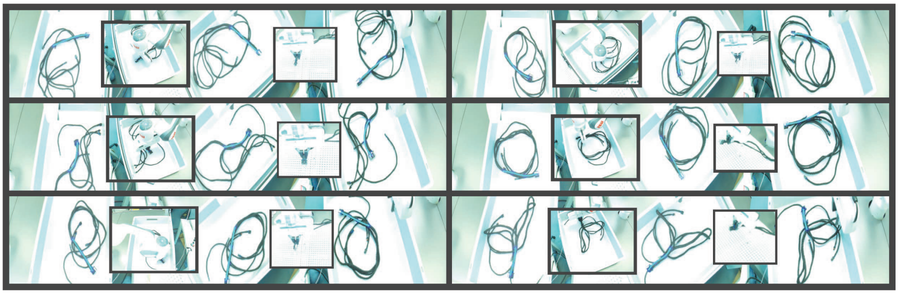


Fig. 11. Six exemplary images of the proposed robotic wiring harness bin picking application. The resulting grasps and the transfer points' final positions are shown in one trial.

Inside one trial there are the three spline predictions from the three cameras, an image right after the grasping and an image showing the connector in a fictive transfer point for further manipulation, which is not part of this paper. The quantitative results of the experiment are shown in Table 2.

Table 2. Experiment results out of 50 trials. If robotic errors aren't accounted for, the overall success rate is 91%.

Outcome	Count	Robot Kinematic Singularities	Collision with Box	Wrong Estimation Wrong Grasp
Successful	41 (82%)	-	-	-
Failed	9 (18%)	1 (2%)	5 (10%)	3 (6%)

The most interesting parts are the failures, explained in the following. Robotic singularities and collisions with the box are not directly addressed in this paper and therefore not part of any further investigation. The interesting parts concern the wrong estimation and wrong grasp section. Three different failure scenarios occurred. Once, the grasp was unsuccessful as the estimated grasp point was too close to the connector, which can be detected using the resulting gripper width, and ended up in a failed attempt. In another trial, the estimated depth was too large, such that the robot grasped an underlying cable as well. The third failure occurred due to a failed spline prediction in one image, detected by a low certainty score of an average of 1.23 for the spline prediction, compared to an average of around 5–6 for the successful spline predictions.

4 Conclusion

This paper presents a robotic bin-picking solution for unstructured wiring harness with minimal initial configuration constraints. It consists of a multi-stereo camera solution and a deep-learning based spline segment estimation of the pickable wiring harness segment. The overall success rate over 50 trials with various configurations is 82%. However, if trials are neglected which have failed due to robotic singularities or collision with the box, the success rate increases to 91%. This paper presents a method for generic intrinsic and extrinsic camera calibration using modified and novel methods. Furthermore, a novel deep-learning architecture with novel spline loss is presented. The proposed system was able to deal with highly overlapped wires. A transfer point was chosen as a final state for the robotic handling, which requires a second robot in the future to grasp the connector and install it into a car door.

The camera system generates extensive, accurate sensor data, of which only a small portion is used for point estimation. A multi-stage low- to high-resolution approach could enhance accuracy. 3D reconstruction algorithms like space carving [29] and shape-from-silhouette [30] could further improve results. For the robot, advanced control with singularity and collision avoidance, and strategies for selecting viable grasping points can be added. Currently, only a single segment of the harness is considered. Future work should address full topology estimation and develop solutions that generalize across different wiring harnesses.

Supplementary information Dataset with camera calibration and validation: *doi.org/10.35097/1n5nqty3c1rxekhk.*
Dataset with annotation, model weights logs: *doi.org/10.18419/darus-4236.*

Acknowledgment. The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the InnovationsCampus Future Mobility (ICM).

References

1. Automotive wiring harness market (2024). https://www.precedenceresearch.com/automotive-wiring-harness-market
2. Nguyen, H.G., Kuhn, M., Franke, J.: Manufacturing automation for automotive wiring harnesses. Procedia CIRP **97**, 379–384 (2021). https://doi.org/10.1016/j.procir.2020.05.254
3. Salunkhe, O., et al.: Review of current status and future directions for collaborative and semi-automated automotive wire harnesses assembly. Procedia CIRP (2023). https://doi.org/10.1016/j.procir.2023.09.061
4. Uhlig, D.: Light Field Imaging for Deflectometry, PhD thesis, Karlsruher Institut für Technologie (KIT) (2023)
5. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992). https://doi.org/10.1109/34.121791
6. Nguyen, T.P., Yoon, J.: A novel vision-based method for 3D profile extraction of wire harness in robotized assembly process. J. Manuf. Syst. **61**, 365–374 (2021). https://doi.org/10.1016/j.jmsy.2021.10.003
7. Kicki, P., et al.: Tell me, what do you see?-interpretable classification of wiring harness branches with deep neural networks. Sensors **21**(13), 4327 (2021). https://doi.org/10.3390/s21134327
8. Guo, J., Zhang, J., Gai, Y., Wu, D., Chen, K.: Visual recognition method for deformable wires in aircrafts assembly based on sequential segmentation and probabilisitic estimation. In: 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), vol. 6, pp. 598–603. IEEE (2022)
9. Zhang, X., Domae, Y., Wan, W., Harada, K.: Learning efficient policies for picking entangled wire harnesses: an approach to industrial bin picking. IEEE Rob. Autom. Lett. **8**(1), 73–80 (2023). https://doi.org/10.1109/lra.2022.3222995
10. Wnuk, M., Zürn, M., Paukner, M., Ulbrich, S., Lechler, A., Verl, A.: Case study on localization for robotic wire harness installation. In: Kiefl, N., Wulle, F., Ackermann, C., Holder, D. (eds,) Advances in Automotive Production Technology – Towards Software-Defined Manufacturing and Resilient Supply Chains, pp. 333–343. Springer, Cham (2023)
11. Jiang, X., Koo, K., Kikuchi, K., Konno, A., Uchiyama, M.: Robotized assembly of a wire harness in car production line. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 490–495 (2010). https://doi.org/10.1109/IROS.2010.5653133
12. Žagar, B.L., et al.: Copy and paste augmentation for deformable wiring harness bags segmentation. In: 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 721–726 (2023). https://doi.org/10.1109/AIM46323.2023.10196168
13. Zürn, M., Wnuk, M., Lechler, A., Verl, A.: Topology matching of branched deformable linear objects. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 7097–7103 (2023). https://doi.org/10.1109/ICRA48891.2023.10161483
14. Wang, H., et al.: Overview of computer vision techniques in robotized wire harness assembly: current state and future opportunities. Procedia CIRP **120**, 1071–1076 (2023)
15. Moons, T.: 3D reconstruction from multiple images part 1: principles. Found. Trends Comput. Graph. Vis. **4**(4), 287–404 (2010). https://doi.org/10.1561/0600000007

16. Raspberry Pi: Raspberry Pi Documentation - Camera. https://www.raspberrypi.com/documentation/accessories/camera.html
17. Zhang, Y.-J.: 3D Computer Vision, 1st edn. Springer, Singapore (2024). https://doi.org/10.1007/978-981-19-7603-2
18. Grossberg, M.D., Nayar, S.K.: A general imaging model and a method for finding its parameters. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, pp. 108–115 (2001). https://doi.org/10.1109/ICCV.2001.937611
19. Hezel, D.: Design and programming of a setup for automated camera calibration, Bachelor's thesis, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany (2024)
20. Bradski, G.: The OpenCV Library. Dr. Dobb's J. Softw. Tools (2000)
21. Schneider, P.J., Eberly, D.H.: Geometric Tools for Computer Graphics. The Morgan Kaufmann Series in Computer Graphics. Elsevier Science (2003)
22. Murugan, A.: Iterative closest point (ICP) algorithm in python (2020). https://github.com/iitaakash/icp_python/tree/master
23. Liu, Z., et al.: Swin transformer v2: scaling up capacity and resolution. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
24. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106
25. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
26. Nibali, A., He, Z., Morgan, S., Prendergast, L.A.: Numerical coordinate regression with convolutional neural networks. arXiv preprint arXiv:1801.07372 (2018)
27. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework (2019)
28. Virtanen, P., et al.: Fundamental algorithms for scientific computing in Python. SciPy 1.0. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
29. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. Int. J. Comput. Vis. **38**, 199–218 (2000)
30. Szeliski, R.: Rapid octree construction from image sequences. CVGIP Image Underst. **58**(1), 23–32 (1993)