

Robot Safety Monitoring using Programmable Light Curtains

Karnik Ram[†], Shobhit Aggarwal[†], Robert Tamburo[†], Siddharth Ancha[‡], Srinivasa Narasimhan[†]

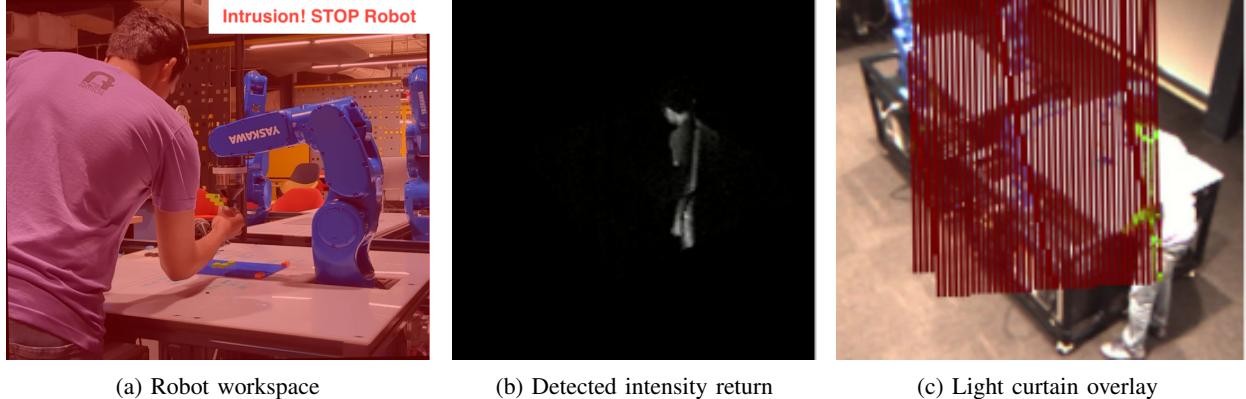


Fig. 1: **Overview.** (a) A human worker walks into the robot workspace to adjust the work-piece on its end-effector. (b) The programmable light curtain device detects the worker on the intensity image and commands the robot to stop. (b) The projected curtain forms a tight convex hull around the two robots for safety monitoring and adapts to their motion in real-time. The intrusion detected through the curtain is shown in green.

Abstract—As factories continue to evolve into collaborative spaces, with multiple robots working together with human supervisors in the loop, the problem of ensuring safety for all actors involved becomes critical. Presently, laser-based light curtain sensors are widely used in factories for safety monitoring. While these sensors have high accuracy standards, they are difficult to reconfigure and can only monitor a fixed user-defined region of space, and are typically expensive. We leverage a recently-developed controllable depth sensor, *Programmable Light Curtains*, for building an inexpensive and flexible real-time safety monitoring system. This system can project tight dynamic safety envelopes that enables fence-less human-robot collaboration, can scale to monitor multiple robots with few sensors, and by utilizing each sensor as a 3D depth sensor the system can also monitor the entire scene within its field of view. We deploy this system in a real testbed environment with four robot arms and demonstrate its capabilities as a powerful safety monitoring solution while being significantly cheaper and not compromising on accuracy.

I. INTRODUCTION

Despite several advances in factory automation using robots, human operators are still required in the assembly process for operations that involve unpredictable behavior (eg. fabric, rubber) or fine-tuned sensitivity (eg. electronics assembly). To enable seamless collaboration between humans and robots in assembly processes, it is necessary to develop robust safety systems that safeguard the collaborative environment and optimize the overall production cycle. Such

systems should incorporate advanced sensors, intelligent algorithms, and clear communication protocols.

Current safety systems can be broadly categorized into two approaches – fixed barriers and collision avoidance [1, 2]. Fixed barriers are physical barriers, such as fences, light curtains, and ToF sensors, demarcate the robot’s workspace and isolate it from human activity. While simple to implement and computationally efficient, they limit collaborative work and offer low flexibility for shared spaces. Approaches based on collision avoidance utilize vision systems such as cameras [3], lasers [4], and 3D time-of-flight sensors [5, 6] to dynamically adapt the robot’s behavior based on its surroundings. This enables a higher degree of collaboration and dynamic shared spaces but typically require multiple devices and powerful computational resources (CPUs and GPUs) for processing the raw sensor streams.

Programmable light curtains (PLC) [7, 8] is a recently developed controllable 3D sensor that images points only along a user-specified 2D surface (called a ‘curtain’). These curtains, which are projected by a steerable sheet laser, can be adapted in real-time to image desired objects or regions of interest in high resolution. The sensor has a working range of upto 30m, is robust even in the presence of ambient light, and is very inexpensive to build (< \$1000). Previously, PLC have been shown to be useful for various tasks in robotics such as object detection [9], depth estimation [10], navigation [11], and safety envelope estimation [12]. In this work, we use PLC to build a flexible safety monitoring system for industrial robots. PLC are used to detect the presence or absence of objects around robots by projecting ‘safety curtains’ that adapt to the configuration of robots, as well as quickly detect the presence of objects in the

[†]Robotics Institute at Carnegie Mellon University, Pittsburgh, USA. [‡]Massachusetts Institute of Technology, Cambridge, USA. Correspondence email: karnikram@gmail.com, {shobhita, srinivas}@andrew.cmu.edu.

Project webpage: <https://rebrand.ly/plc-safety>

surrounding scene by using ‘random curtains’. The system is very inexpensive compared to traditional laser-based safety systems and easily scales to many robots with only few PLC sensors.

The remainder of the paper is structured as follows. We begin by describing the PLC sensor and its working principle in [Sec. II](#). We then describe our approach to implement a safety monitoring system using PLC sensors in [Sec. III](#). We implement such a system in an experimental testbed setup which we present in [Sec. IV](#), before concluding with a discussion in [Sec. V](#).

II. BACKGROUND

A. Working Principle of Programmable Light Curtains

Programmable light curtains (PLC) [7, 8] are a recently-developed controllable depth sensor that images only along a specified vertically-ruled 2D surface in the environment. The sensor contains two main components: a rolling-shutter near-infrared (NIR) camera and a rotating sheet NIR laser (see [Fig. 3a](#)). The rolling-shutter camera activates one pixel column at a time, and we refer to the top-down projection of the imaging plane corresponding to each pixel column as a “camera ray” (see [Fig. 3b](#)). A 2D control point is selected on each camera ray (shown as gray and green circles). A controllable galvo-mirror rotates the laser light sheet in sync with the rolling shutter camera to point it at the control point corresponding to the currently active pixel column. 3D scene points that lie at the surface of this intersection between the projected light sheet and the image plane (called ‘curtain’) get imaged by the camera (shown as green circles). The set of control points completely determine the shape of the light curtain and subsequently the objects that are imaged. These control points form the input to the PLC and can be adjusted at the camera’s frame rate to image different surfaces in each frame. Additionally, an RGB camera is used as a ‘helper camera’ to visualize the light curtains as shown in [Fig. 1](#). Please refer to [8] for more details about the PLC such as its hardware, performance under various conditions etc.

III. METHOD

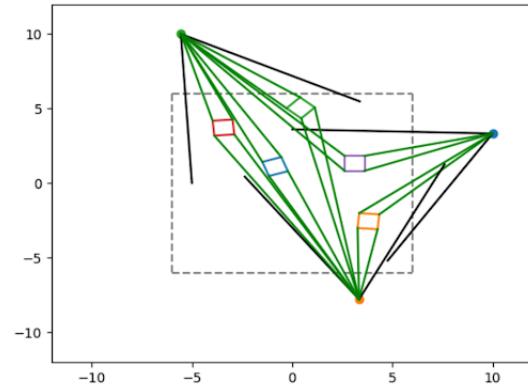
A. Instrumentation

For a given layout of robot arms on a factory floor, we begin by determining the positions and orientations of the PLCs required to safely envelope all the robots. We design an algorithm to do this computationally through exhaustive random search.

We represent each robot and PLC in 2D as a 4-sided polygon and point respectively. The four corners of each robot polygon are provided as input to the algorithm and remain fixed, while the 2D pose (x, y, θ) of each PLC is to be estimated. For each PLC, we randomly sample a 2D pose within the dimensions of the floor space and determine the robots within its field-of-view. For every robot within its field-of-view, we compute the angle subtended by the closest visible edge of the robot at the PLC. We only consider a robot’s edge if it has not already been observed by another PLC. The sum of angles subtended by these unique visible

edges at every PLC is used as a measure of coverage quality for this configuration of sampled PLC poses. We repeat this process for a total of N iterations and return the configuration of PLCs which amounts to the highest sum of subtended angles. The number of PLCs required to envelope all robots can also be determined by simply running the algorithm for an increasing number of PLCs, starting from 2, until all the visible edges of the robots are observed.

The result of this algorithm for a test case of five robot arms, each represented as a square polygon, and three PLC sensors is shown in [Fig. 4](#). The algorithm was run for 1 million iterations which took 1 hour to complete.



[Fig. 4](#): Estimated configuration of PLCs (represented as dots) for a $12 \times 12 \text{ m}^2$ example layout containing five robots (represented as squares). The algorithm returns the configuration that maximizes the sum of the angles subtended by the robots at the PLCs (rays denoted by —) over many random trials.

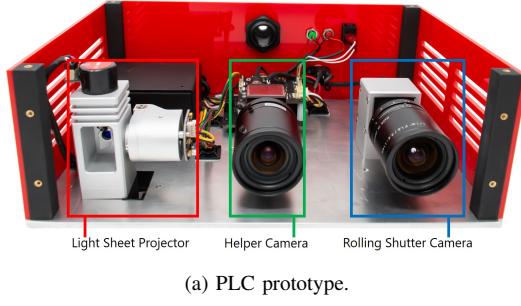
B. Safety Curtain Design

We design safety curtains from each PLC based on the known poses of the robot arms in the scene. To do this, we first obtain the joint positions provided by the controller of each robot arm and project them to a 2D plane in the frame of reference of the closest PLC. Since the number of joint positions are usually small (6-8), we also use the positions of some pre-defined virtual joints from the shape of the robot. We then compute a 2D convex hull containing all of these points. Finally, to compute the N control points of the curtain we do 2D ray-tracing for each of the N camera rays of the PLC on this 2D plane containing the convex hull. The points of intersections of each ray with the hull determine the control points and hence the shape of the safety curtain. A small additional offset is added to the control points to ensure the curtains don’t intersect with the robots.

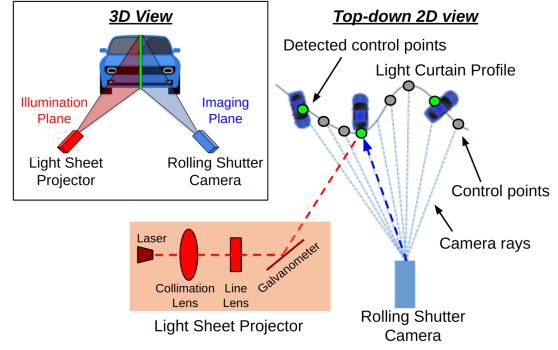
This computation is repeated for every frame as the robots move around their workspace, ensuring that the safety curtains are dynamic and closely track the robots’ motion. For cases when the joint positions are not completely available, random curtains can be used to determine the robot positions and subsequently tracked as done in [12].

C. Intrusion Detection

If an object intersects with the safety curtain and there is an intensity return on any PLC, we determine which robot(s)



(a) PLC prototype.



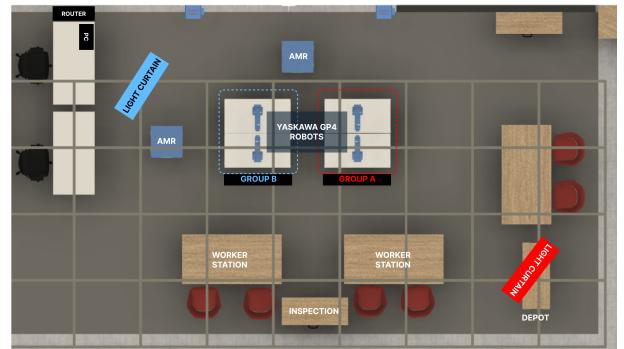
(b) Working principle of PLC.

Fig. 3: (a) Our PLC prototype consists of a near-infrared (NIR) light sheet laser mounted on a rotating galvomirror, an NIR rolling shutter camera, and an RGB helper camera for visualization. (b) The light sheet laser rotates in sync with the rays of the rolling shutter camera to selectively image fixed points along a surface. Figures adapted from [8, 9].

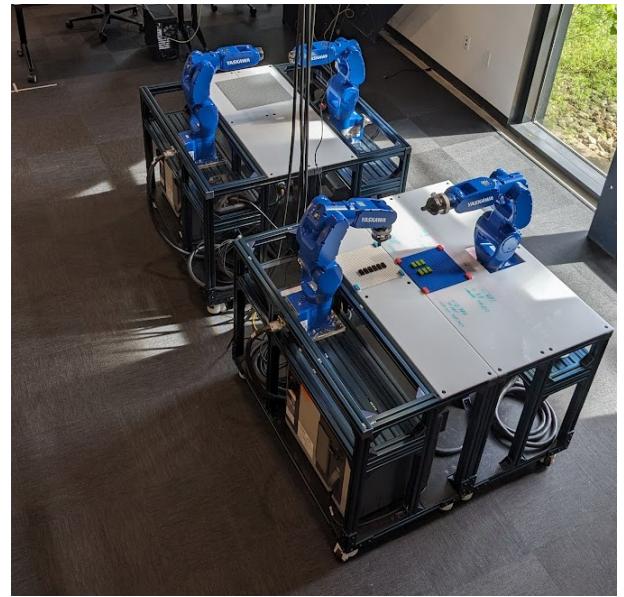
it corresponds to based on the location in the returned image (or distance) and stop the robot immediately. Interference from other sensors may also occasionally cause intensity returns. Since these interference returns can usually last between 2 to 4 frames, we stop a robot only if there is a continuous intensity return for more than 4 consecutive frames. In situations where utmost safety is prioritized over efficiency, the robots can be temporarily stopped whenever an intensity return is received, regardless of whether it originates from an object or interference. Additionally, since these interference returns have a distinct unnatural pattern in the image, we can also train a small on-device neural network to determine if the intensity return is caused by interference or an object.

D. Scene Reconstruction

In addition to safety curtains, we also use each sensor as a complete 3D depth sensor by sweeping the scene with fixed-shape (or random) curtains and merging using their intensity returns for full depth reconstruction. Each intensity return has an associated depth map as each pixel corresponds to a known camera-ray – laser sheet intersection by construction. These intensity returns from a full sweep can then be merged together into a single intensity image by storing the maximum intensity value at each pixel location across the sweep of images. This merged intensity image is then backprojected to a full 3D point cloud using the known depth value and the camera’s intrinsics. Since the PLC can project multiple light curtains at the camera’s frame rate of 45–60 Hz, these random curtains can be interleaved with the safety curtains during operation. The point clouds generated from each sensor are sent to a centralized workstation where they are processed and registered together using the iterative closest point (ICP) algorithm [13]. The resulting point clouds could be further used to detect other mobile robots or people in the workspace and accordingly compute additional curtains to track them for safety.



(a) *Testbed layout*. Four robot arms are arranged together in pairs on workstations that are x m from the ground. Two PLCs (in blue and red) are mounted on an 80/20 grid from the ceiling.



(b) The four Yaskawa® robot arms as observed by the red PLC.

Fig. 5: Testbed area used in our experiments.

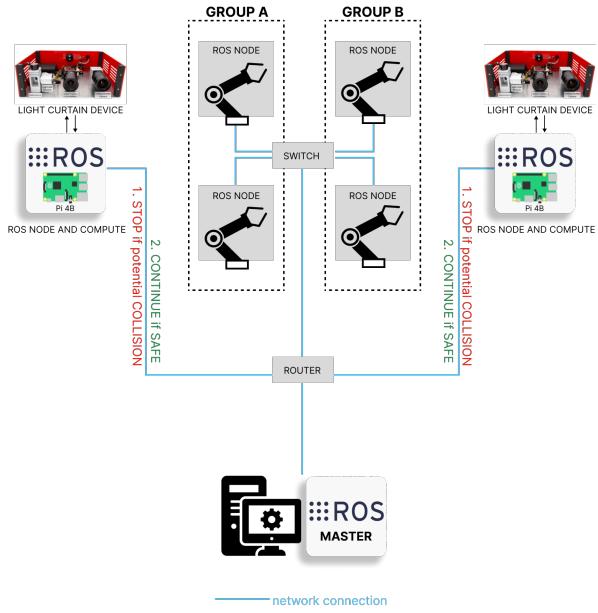


Fig. 6: Network setup used in the testbed. The PLC sensors are connected to the robots over ROS using ethernet cables. The robots are monitored and controlled for safety directly by the PLC sensors. The workstation is used for visualization and additional control.

IV. EXPERIMENTS

A. Setup

Testbed. We demonstrate the capabilities of our safety monitoring system in a manufacturing testbed as shown in Fig. 5. The testbed consists of two pairs of Yasakawa® GP4 robotic arms arranged in a square layout, with each pair collaborating to assemble and disassemble a Lego® model. Each robot is equipped its own control PC which is also connected to a central workstation over ROS. There are two autonomous mobile robots (AMRs) that move material between the arm robots and the worker stations. We have two downward-facing PLCs mounted on a 80/20 grid that is attached to the ceiling at a height of 3.35m from the ground. Each PLC has a Raspberry-Pi® 4 onboard which runs all the code for designing and imaging the light curtains. The total dimensions of the testbed are $9.3 \times 5.9 \text{ m}^2$ and all the communications are carried over ethernet to avoid latencies. The pose between each robot and PLC is calibrated with an eye-on-base calibration procedure using Apriltags.

Eye Safety. Eye safety requirements limit the maximum power at which the laser on the PLC can be operated [14, 15], which limits the maximum range of the sensor. In our testbed, we operate the laser at 30% of its maximum power (1W), which corresponds to a maximum eye-safe distance of 16.80 cm. This distance is well beyond the reach of people during regular operation. For further precaution, we also monitor the output of the sensor to ensure that the galvomirror is rotating and that the laser's power level is maintained. To avoid introducing additional circuitry into the prototype sensor, we achieve this by attaching a white planar sheet as an extension to the sensor at a short distance within its field-of-view (shown in Fig. 7). This sheet is scanned by the PLC every second and if the intensity return from

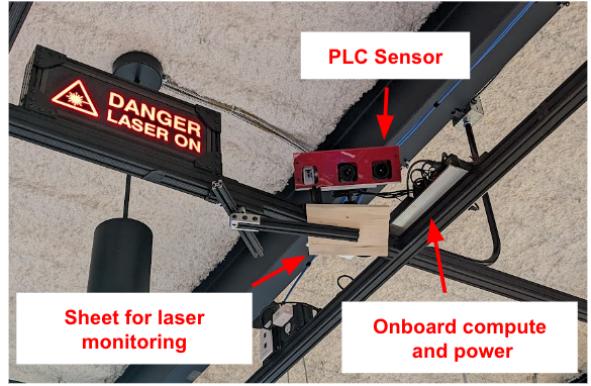


Fig. 7: Red PLC sensor mounted in the testbed with on-board compute and power, and a laser safety mechanism.

it does not match the expected return, the mirror is deemed malfunctioning and the sensor is immediately turned off. The returned intensity values are also monitored to ensure that the laser's output power is constant and is not fluctuating.

B. Results

Instrumentation. The two PLCs in our testbed are positioned according to results from our algorithm described in Sec. III-A. Unlike regular laser-based safety sensors that typically can monitor only one robot each, our two PLCs can monitor all four robots as well as monitor the full area in the testbed surrounding them for people and other mobile robots.

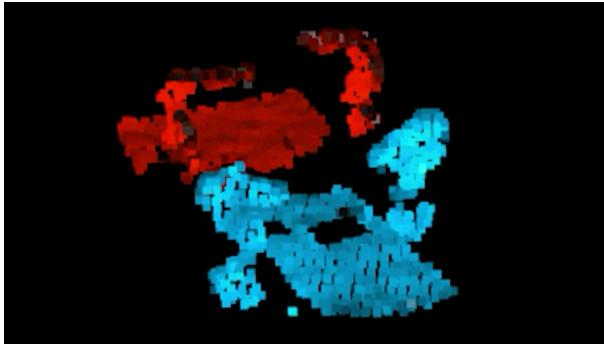
Safety Curtains. The blue and red PLCs receive the joint positions of the group A and group B robots respectively over ROS at 100 Hz. These positions are used the sensors to compute in real-time the safety curtains required to envelope the robots at their current pose. These curtains are recomputed for every frame and projected to follow the robots' movement, ensuring that the robots are always enveloped. This tracking behaviour of the sensors is visualized in the video accompanying the paper. Unlike regular laser-based safety systems which typically block the area around the robots, our safety curtains follow the robots more tightly which enables fence-less human-robot collaboration.

Intrusion Detection. When there's an intersection of the safety curtain with a person or object, laser reflects off the object and is captured directly by the NIR camera. These intrusions are detected at the optical level by design and are hence very accurate. Visualizations of such a detection are shown in Fig. 1. Objects of all sizes from a full-sized human to small parts of the hand can be quickly detected. For objects that are of dark colors, the laser operating power would need to be increased to detect them as they tend to absorb some of the light. When an intrusion is detected the corresponding robot is immediately commanded to stop. The robot resumes its motion once the intruding object disappears.

3D Reconstruction. Apart from projecting safety curtains, the PLC sensors also project fixed-shape curtains periodically to obtain a full 3D reconstruction of the testbed area. For this experiment, these fixed-shape curtains are planar curtains that sweep the testbed area with a distance of 0.01 cm between each planar curtain. The returned intensity images



(a) Merged intensity returns from the blue PLC.



(b) Merged point cloud from the blue and red PLCs.

Fig. 8: *Pointcloud reconstruction*. The testbed area is scanned using planar curtains by the blue and red PLCs, and their intensity returns are processed and registered together to obtain a reconstruction in 3D.

are merged into a single intensity image (shown in Fig. 8a) and its corresponding pointcloud reconstruction is obtained as described in Sec. III-D. The pointclouds from both PLCs are then filtered and registered together using ICP on the central workstation with the calibrated poses of the PLCs used as initialization. The resulting pointcloud for the four robots is visualized in Fig. 8b.

Latency. The update rate of the safety curtains that are designed and imaged on the sensor (RPI-4) is 7 Hz. The computationally expensive step is the design of each safety curtain based on ray-hull intersections, which implemented as unoptimized Python code takes 130 ms on average. The update rate of the planar curtains used for 3D reconstruction is 22 Hz, and is the same for random curtains as well.

V. DISCUSSION

We presented a new inexpensive safety monitoring system using programmable light curtains. The system is flexible

and can be used for monitoring arbitrary configurations of robots as well as the full 3D scene, while easily scaling to many robots. Future work would be to incorporate trajectory forecasting of the robots as well as the objects into the pipeline to avoid recomputing safety curtains for every frame. Such forecasting would fully leverage PLC’s active sensing capabilities and enable new active safety monitoring systems that are able to pre-empt dangerous situations instead of responding passively. Current limitations of the system include inaccurate depth sensing at large distances (greater than 8 m) due to curtain thickness [8] which can be alleviated using probabilistic depth models [10], and less sensitivity to dark colored objects which can be alleviated using online laser power calibration algorithms.

REFERENCES

- [1] S. Robla-Gómez, V. M. Becerra *et al.*, “Working together: A review on safe human-robot collaboration in industrial environments,” *Ieee Access*, vol. 5, pp. 26 754–26 773, 2017.
- [2] A. Buerkle, W. Eaton *et al.*, “Towards industrial robots as a service (iraas): Flexibility, usability, safety and business models,” *Robotics and Computer-Integrated Manufacturing*, vol. 81, p. 102484, 2023.
- [3] M. Ozkahraman, C. Yilmaz, and H. Livatayali, “Design and validation of a camera-based safety system for fenceless robotic work cells,” *Applied Sciences*, vol. 11, no. 24, p. 11679, 2021.
- [4] K. Zheng, F. Wu, and X. Chen, “Laser-based people detection and obstacle avoidance for a hospital transport robot,” *Sensors*, vol. 21, no. 3, p. 961, 2021.
- [5] S. Pasinetti, C. Nuzzi *et al.*, “Development and characterization of a safety system for robotic cells based on multiple time of flight (tof) cameras and point cloud analysis,” in *2018 Workshop on Metrology for Industry 4.0 and IoT*. IEEE, 2018, pp. 1–6.
- [6] Intel, “Using intel® fpgas in veo freemove* system to enable safe, dynamic, and flexible human-robot collaboration,” 2020. [Online]. Available: <https://www.intel.com/content/dam/www/central-libraries/us/en/documents/fpgas-in-veo-freemove-case-study.pdf>
- [7] J. Wang, J. Bartels *et al.*, “Programmable triangulation light curtains,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.
- [8] J. R. Bartels, J. Wang *et al.*, “Agile depth sensing using triangulation light curtains,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7900–7908.
- [9] S. Ancha, Y. Raaj *et al.*, “Active perception using light curtains for autonomous driving,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 751–766.
- [10] Y. Raaj, S. Ancha *et al.*, “Exploiting & refining depth distributions with triangulation light curtains,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7434–7442.
- [11] S. Ancha, G. Pathak *et al.*, “Active velocity estimation using light curtains via self-supervised multi-armed bandits,” in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.
- [12] S. Ancha, G. Pathak, S. Narasimhan, and D. Held, “Active Safety Envelopes using Light Curtains with Probabilistic Guarantees,” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [13] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.
- [14] S. Acharya, J. R. Bartels *et al.*, “Epipolar time-of-flight imaging,” *ACM Trans. Graph.*, vol. 36, no. 4, jul 2017. [Online]. Available: <https://doi.org/10.1145/3072959.3073686>
- [15] A. N. S. Institute, *American National Standard for Safe Use of Lasers*, 2014. [Online]. Available: <https://webstore.ansi.org/standards/lia/ansiz1362014>