# Model-based Multiple Object Tracking Using Capacitive Proximity Sensors

Siyuan Huang[1], Hosam Alagi[1] and Björn Hein[1,2]

***Abstract*— In this work, an object tracking method on capacitive proximity sensor is presented. Arranged as a vector and installed in the sidewalls of a work table with a robot, these sensors are able to detect proximity events in the near field of the workspace, including human approaching the table or interacting with the robot. Considering the low spatial resolution of the sensors, a preprocessing method was implemented and the data for tracking was prepared. The tracking method is based on a Kalman-Filter, where the data association and occlusion problem are discussed. The Methods implemented in MATLAB show that the performance of this proposed tracking using capacitive proximity sensor array for safe human-robot interaction.**

## I. INTRODUCTION

Safe HRI imply that the robot system is able to interact with the environment and thus guarantee safety in every moment, when humans and robot are sharing the same workspace [1]. For the perception of the environment, many sensors are required, such as camera, laser, etc. However, the traditional camera-based or laser-based solutions are problematic, since they are encumbered by occlusions, light influence [2] and reflection effect. In contrast, the proximity sensors attached on the robot's surface enable the exploration and modeling of the close environment [3].

The presented work is a continuation of our previous work [4], where a modular capacitive tactile proximity sensor was proposed for the applications in robotics. In this work, we attached these sensor modules in the sidewalls of a worktable and introduce methods from computer vision to process the signal. We also implemented a multiple object tracking method based on Kalman Filter in order to gain more information from the time series.

## II. RELATED WORK

As stated in the introduction section, the proximity sensors have the ability to complement the robot's perception in its near field, which could ensure a safe interaction between human and robot. In [5], proximity sensors are used to gain information about the object's position and primitive shapes, and based on these useful information, the robot can efficiently execute the grasping task. A network consisting of proximity sensors is able to locate the objects placed on the network, as described in the work [6]. In [2], the concepts for object tracking based on tactile proximity sensor are presented. In [7] and Bosch APAS[8], a proximity skin sensor attached to the surface of the robot enables the detection of the object's position and its approximate shape. The work [9] proved that the proximity sensors could offer the information about the material of the approaching object.

This paper is structured as following: firstly, we give a brief introduction about the sensor array used in this work, we define two different working states of the tracked objects "human". Later on, we represent how computer vision methods are used to detect the objects and explain the implementation of Kalman-Filter for tracking. The evaluation results are then shown for two humans with occlusion case.

## III. SYSTEM DESCRIPTION

In this section we present the basic characteristic of the sensor and then the concrete algorithm implementation, including preprocessing and tracking methods.

### A. Sensor Array

The structure and working principle for proximity sensor module shown in Fig. 1 was presented in [4]. Each module can address 8 electrodes. The size of the electrodes can vary depending on the application; the bigger the size is, the higher the measurement range will be. In this work, we equipped each wall of the worktable with a stripe of 8 electrodes, each has the size of $10 \times 10\ cm^2$ a measurement range of 15 $cm$. Those are driven by 3 sensor modules via a coaxial cable, as shown in Fig. 2.

### B. Working States of Human

Unlike the works [7] and [8], where the proximity skin sensor is attached to the robot's surface, we install the sensors alongside the sidewall of the worktable, as shown in Fig. 2.

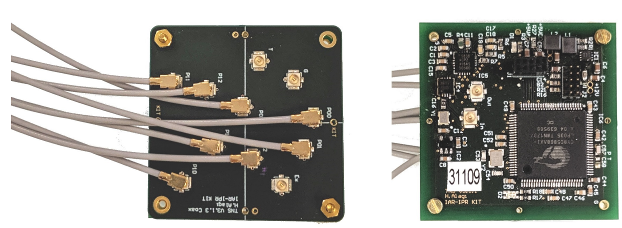


Fig. 1: CTPS module. Top and bottom side

[1]Authors are with Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics Lab (IAR-IPR), Karlsruhe Institute of Technology siyuan.huang@student.kit.edu, hosam.alagi@kit.edu

[2]Björn Hein is with the Karlsruhe University of Applied Science bjoern.hein@hs-karlsruhe.de

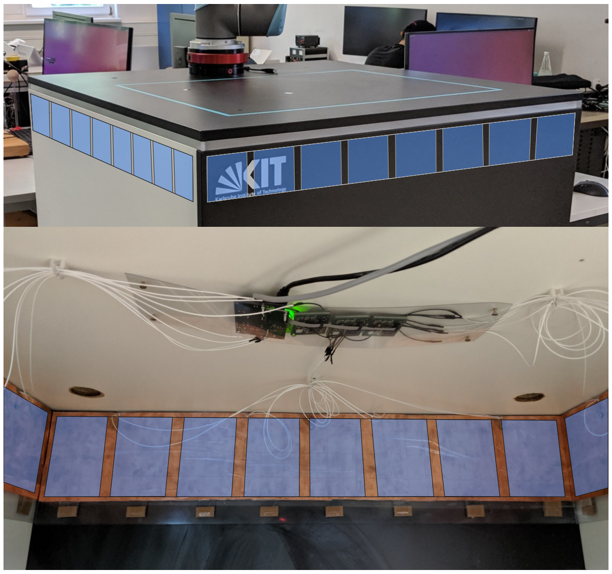



Fig. 2: Capacitive proximity sensors installed in the table

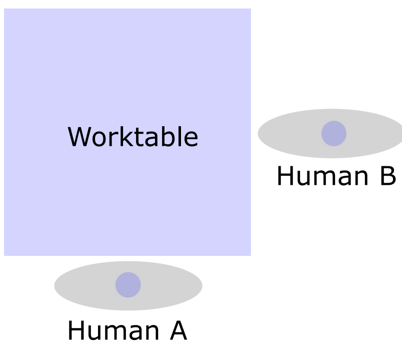



Fig. 3: Working states of the human

In this way, we no longer focus on the direct interaction between human and robot, instead, the working states of human are more interesting. Based on this, the robot could adapt dynamically to achieve a safe HRI. Also, the model developed in this work will not highly depend on specific robot, supporting the transparency and generalization.

Considering the highly-structured working situation in a factory, we could roughly divide the working states of human into two classes, static and dynamic. The static class describes usually the standing pose of the operators, when they interact with the robot, as *Human A* in Fig. 3. In contrast, the dynamic class describes the walking pose, where the operators are passing by the worktable, as *Human B*.

Accordingly, these two classes have different characteristics as shown in Tab. I, which support higher tracking robustness.

### C. Object Detection

As the data of the sensor array could be taken as a one-dimension planar image, we could introduce methods from computer vision to detect moving objects in the proximity of the worktable. In the following section, we explain the methods used for the tracking.

TABLE I: Characteristics of working states

| | Static | Dynamic |
|---|---|---|
| Projected Area | Big | Small |
| Velocity | Low | High |
| Illustration in Pic. 3 | Human A | Human B |

**Background subtraction.** The background used here is the zero mean values of first hundred frames after sensor initialization. And the subtraction can be expressed as follows:

$$BS_t(x) = \begin{cases} 0 & \textit{if}\,(I_t(x) - B(x)) < \tau_{BS} \\ 1 & \textit{if}\,(I_t(x) - B(x)) \geq \tau_{BS} \end{cases} \tag{1}$$

Where $I_t(x)$ is the input sensor values at step $t$, $BS(x)$ is the the background signals and $\tau_{BS}$ is the threshold value. After this process, we assign the areas, where an object exists, to 1, and the other areas to 0. In this way, the sensor values are changed into a binary vector.

**Segmentation.** In this process, the *Connected Component Labeling* algorithm is applied. Taking the binary vector as an input, the output would be a symbolic vector, where the labels are assigned, identifying the unique connected component. Also, with a prior knowledge about the human body, e.g. the minimum projection area should bigger than 15cm, even in dynamic states, we can make a verification to filter out the noise, which is as follows:

$$\begin{cases} Human & \textit{if}\ C_i > \tau_c \\ Noise & Otherwise \end{cases} \tag{2}$$

Where $C_i$ is the component area of $i^{th}$ detected operator, and $\tau_c$ is the threshold value.

### D. Tracking

Compared with the vision sensors which could offer us sufficient information, the spatial resolution of the proximity sensors is quite low and it's quite difficult to distinguish different objects when the sensor array only use a single snapshot. In order to make the sensor array able to find the right correspondence when multiple objects appear, as well as to smooth the noise positions and the trajectories, it is essential to gain more information from time series [2]. In this work, we use the Kalman-Filter.

**System model.** In this work, the same number of trackers as the detected objects are deployed in order to estimate the objects state. Each Kalman Filter tracker is configured as follows:

$$\begin{aligned} x_k &= Ax_{k-1} + \omega_k \\ z_k &= Hx_k + v_k \end{aligned} \tag{3}$$

where the $\omega_k$ and $v_k$ are the Gaussian noise with corresponding error covariances $Q_k$ and $R_k$.

The objects states are described with location $x$, length $l$

of the bounding box that presents the detected object and velocity $v_x$. The state vector could be written as:

$$x = \begin{bmatrix} x & l & v_x \end{bmatrix}^T \tag{4}$$

And the state transition matrix $A$ and measurement matrix $H$ could be represented as:

$$A = \begin{bmatrix} 1 & 0 & T \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

where, T is the time interval between two adjacent sampling frame.

**Data association.** In the multiple object tracking scenario, we obtain several measurements through detection. In order to assign the correct measurement to the corresponding object tracker, we use the method *Intersection over Union (IoU)* that describes the similarity of the sample sets.

In this work, we define a cost of assignment as described in equ. (6). The smaller the cost is, the higher probability of the assignment being true.

$$Cost = \begin{cases} 1 - IoU & if\, IoU > \tau \\ 1 + punishCost & others \end{cases} \tag{6}$$

where $\tau$ is the similarity threshold, and the $IoU$ could be calculated with (7),

$$IoU = \frac{area(A \bigcap B)}{area(A \bigcup B)} \tag{7}$$

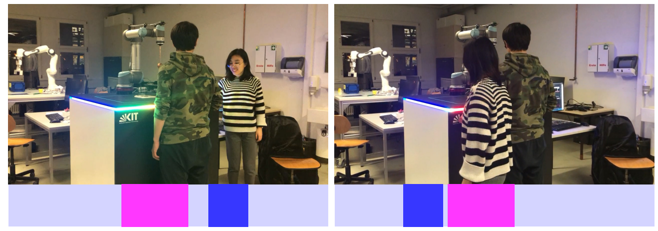
where A represents the detected object area and B represents the predicted object area.

In this way, we get an $M * N$ cost matrix, where $M$ represents the trackers' number, and N for measurement. At that point, the assignment problem turns into a cost minimization problem, which could be solved with the Hungarian algorithm.

## IV. EXPERIMENTAL VALIDATION

In this part, we tested our proposed framework with the worktable, shown in Fig. 2. For the test scenario, we consider the different working states of human, which means one person is standing alongside the worktable while the other is walking by, as shown in Fig. 4a.The tracking result of this framework can be seen in Fig. 4b, where the heliotrope rectangle represents the standing person and the blue one represents the person passing-by.

The framework could distinguish different objects even if a occlusion occurs. One person was standing about 10 $cm$ away from the worktable and the other was working around the worktable with about 1.5 $m/s$ and about 15 $cm$ away. The width of the tracks shown in 4 corresponding to the activated electrodes and thus to distance and the pose of the person.



(a) Before occlusion. (b) After occlusion.

Fig. 4: Test scenario in real world and tracking result in MATLAB

## V. CONCLUSIONS

In this work we presented a framework for multiple object tracking using capacitive proximity sensor array, which is able to perceive their near environment. It shows the benefits of using locally installed sensor to cover large areas of a collaborative-workspace. We see the capacitive proximity sensors as an complementary system for cameras, especially in the case on occlusions.

This is a proof of concept work showing the capability of capacitive proximity sensors for tracking tasks. The future work will extend the use case to include multi-human interacting with the worktable and collaborating with the robot. We will show how the capacitive proximity sensor in combination with other sensors could contribute to a safe Human-Robot-Interaction.

## REFERENCES


[1] P. Svarny, Z. Straka, and M. Hoffmann, "Toward safe separation distance monitoring from RGB-D sensors in human-robot interaction," *arXiv:1810.04953 [cs]*, Oct. 2018, arXiv: 1810.04953. [Online]. Available: http://arxiv.org/abs/1810.04953

[2] S. E. Navarro, M. Marufo, Y. Ding, S. Puls, D. Gger, B. Hein, and H. Wrn, "Methods for safe human-robot-interaction using capacitive tactile proximity sensors," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 1149–1154.

[3] Y. Ding, H. Zhang, and U. Thomas, "Capacitive Proximity Sensor Skin for Contactless Material Detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 7179–7184.

[4] H. Alagi, S. E. Navarro, M. Mende, and B. Hein, "A versatile and modular capacitive tactile proximity sensor," in *2016 IEEE Haptics Symposium (HAPTICS)*, Apr. 2016, pp. 290–296.

[5] K. Koyama, Y. Suzuki, A. Ming, and M. Shimojo, "Integrated control of a multi-fingered hand and arm using proximity sensors on the fingertips," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4282–4288.

[6] Y. Suzuki, K. Koyama, A. Ming, and M. Shimojo, "Grasping strategy for moving object using Net-Structure Proximity Sensor and vision sensor," *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1403–1409, 2015.

[7] S. Tsuji and T. Kohama, "A proximity touch screen using mutual capacitance measurement," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2016, pp. 1262–1266.

[8] BOSCH, "APAS Human-robot collaboration." [Online]. Available: https://www.bosch-apas.com/en/human-robot-collaboration/

[9] H. Alagi, A. Heiligl, S. E. Navarro, T. Kroegerl, and B. Hein, "Material Recognition Using a Capacitive Proximity Sensor with Flexible Spatial Resolution," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 6284–6290.