# SKUBA 2024 Team Description Paper

Kanjanapan Sukvichai, Noppanut Thongton, Nutchanon Nonthapiboon, and Phakhaphol Thengchatapunt

Department of Electrical Engineering,
Faculty of Engineering, Kasetsart University
https://www.robotcitizens.org

**Abstract.** This paper is the team description paper of SKUBA for our engagement in the World RoboCup 2024 @Home Social Standard Platform League held in Eindhoven, Netherlands. Our team aspires to participate in the @Home Standard Platform League using the PEPPER robot. The PEPPER robot is renowned for its potent human interaction capabilities and accessible software conducive to developmental learning. The utilization of the PEPPER robot is envisaged to facilitate the augmentation of the team's skills, aligning with the principal research objectives.

## 1 Introduction

Since 2008, the SKUBA team has actively participated in the RoboCup competition, initially engaging in the SSL competition and achieving a notable third-place finish in our debut year (2008). Subsequently, from 2009 to 2012, we secured first place four consecutive times. In 2012, the team shifted our focus to the RoboCup@Home league, culminating in a victory at the 2019 RoboCup@Home Education Competition in Sydney, Australia. We got a fourth-place position in the RoboCup@Home Open Platform League during RoboCup 2022 in Bangkok. In the RoboCup@Home Social Standard Platform 2023 in Bordeaux, our team marked our debut in this league. Comprising undergraduate and graduate students from Kasetsart University's Faculty of Engineering in Thailand, our team is guided by experienced faculty members. It remains dedicated to offering practical solutions in the field of service robots. Looking ahead, we are enthusiastic about our involvement in the World RoboCup2024@Home Social Standard Platform. The overall software is shown in Figure 1.

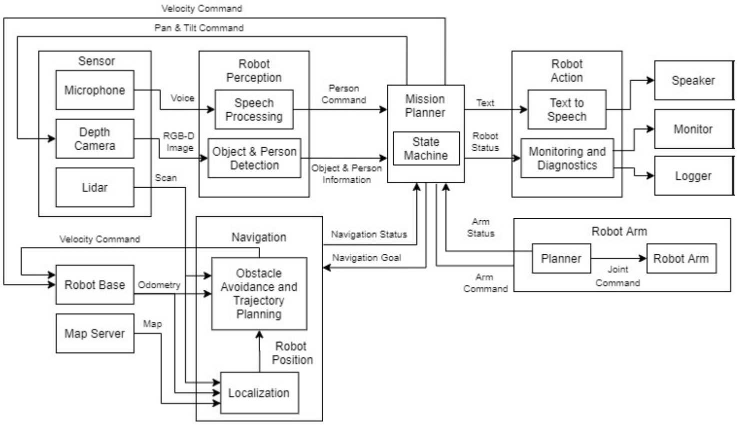


**Fig. 1.** Software Pipeline

## 2 Robot Vision

### 2.1 Object Detection and Recognition

In the previous year, our real-time object detection system utilized YOLOv4-tiny. However, for the current year, we have transitioned to YOLOv7-tiny for object detection and recognition, as shown in Figure 2, developed by Wang, C.Y., Bochkovskiy, A., and Liao, H.Y.M.[10] This newer version optimizes network performance through the implementation of trainable bag-of-freebies methods, effectively reducing parameters and conserving computational resources. We integrate YOLOv7 with the synthetic dataset[4] generator that can generate images with various backgrounds to reduce the time to take a picture and increase the accuracy of the object detection system. The example of output from our synthetic dataset generator is shown in Figure 3.

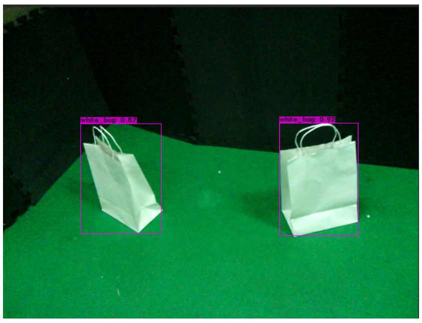


**Fig. 2.** Object Detection

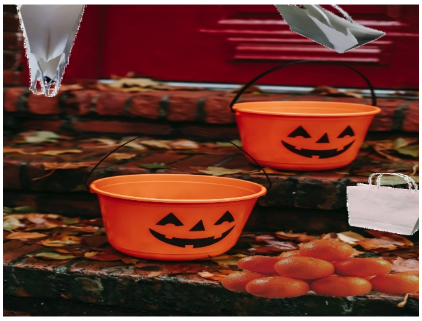


**Fig. 3.** Synthetic data

### 2.2 Pose estimation

OpenPose, an efficient real-time 2D multi-person keypoint detection system, is tailored for streamlined processing on resource-constrained environments, including CPUs[1]. The human action recognition system integrates OpenPose with an LSTM (Long Short-Term Memory) model. The training procedure involves the utilization of keypoints and images to construct a dataset incorporating actions detected by OpenPose. Subsequently, this dataset is employed to train the LSTM model, culminating in the development of a fall detection mode.

### 2.3 Hand gesture recognition

Our hand gesture recognition system is achieved through the integration of MediaPipe[6] and TensorFlow. By leveraging MediaPipe, we extract key landmarks (keypoints) from the hand. These keypoints are then fed into our TensorFlow neural network, which has been specifically designed for the purpose of hand gesture recognition. Our innovative approach combines the robust hand-tracking capabilities of MediaPipe with the computational power and flexibility of TensorFlow. This contributes to the development of an effective and adaptable hand gesture recognition model that is capable of recognizing a wide range of hand gestures. The result from out system as in Figure 4.

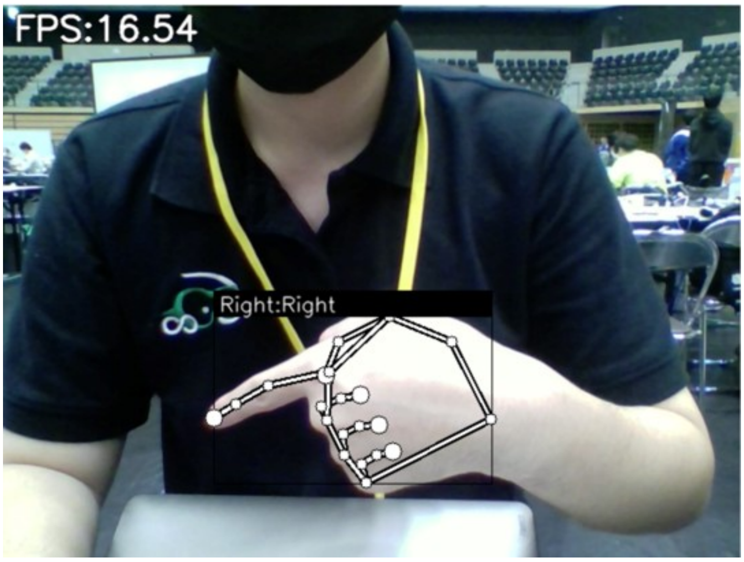


**Fig. 4.** Hand gesture recognition

### 2.4 Human follower

In the pursuit of human tracking, a 3D sensor is employed to discern the spatial coordinates of the person within the point cloud. This involves identifying the region occupied by a person and subsequently computing their precise 3D position. The derived spatial information is then transmitted to the navigation stack, facilitating targeted navigation while mitigating potential obstacles. Concurrently, OpenCV is utilized to ascertain the color of the person's attire, thereby preventing the inadvertent tracking of an incorrect person. As shown in Figure 5, the green area is the data from the person's 3D camera that we use to track the person and find the person's outfit color.

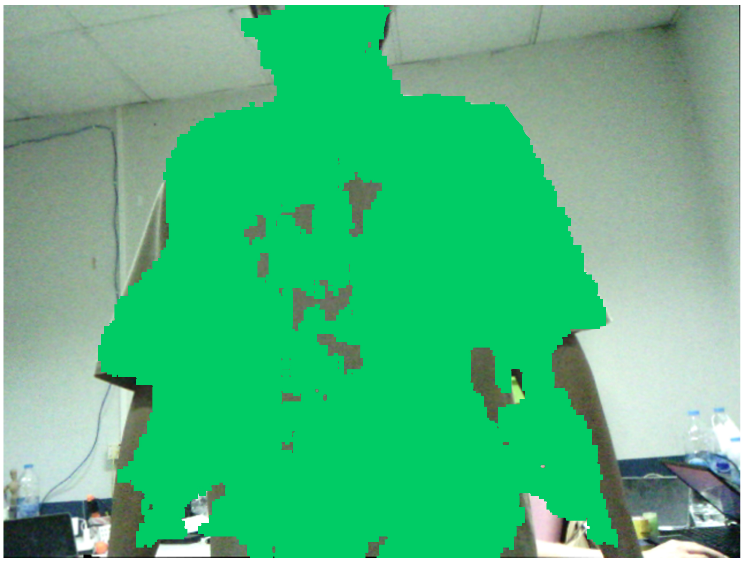
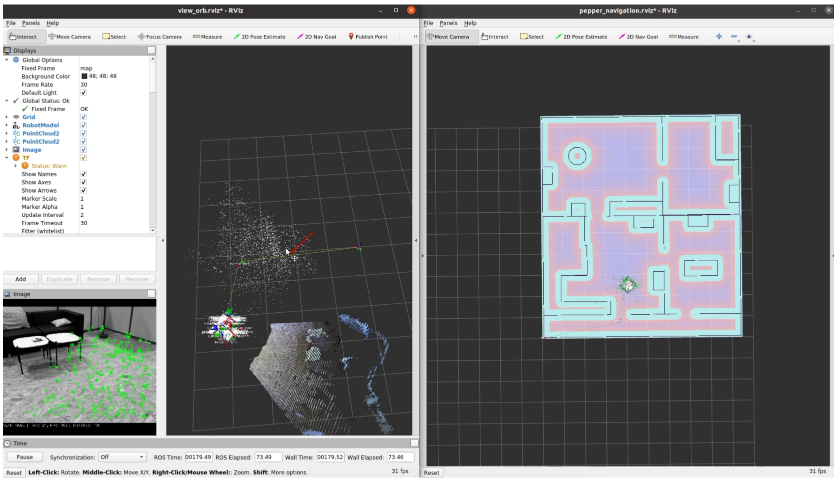


**Fig. 5.** Hand gesture recognition

## 3 Localization and Navigation

### 3.1 Visual simultaneous localization and mapping (vSLAM)

In response to the limitations posed by the sensors on the Pepper robot, we opted for a visual Simultaneous Localization and Mapping (SLAM) approach to address its localization challenges. Leveraging the richness of information inherent in 3D camera data, we employed the ORB-SLAM algorithm with the 3D camera to localize the robot effectively [8]. Subsequently, a fusion process was implemented, integrating the outputs from ORB-SLAM with additional data streams obtained from the robot's encoder and LiDAR sensors. The amalgamation of these diverse sensor inputs was facilitated by applying an Extended Kalman Filter (EKF) by robot localization package [7], contributing to a refined and more accurate estimation of the robot's positional coordinates. This integrated methodology enhances the overall localization precision of the Pepper robot, compensating for the limitations of its sensors through the synergistic exploitation of visual and depth-based sensor information. The output is shown in Figure 6.



```
Global Options
  Fixed Frame          map
  Background Color     48; 48; 48
  Frame Rate           30
  Default Light        ✓
✓ Global Status: Ok
  ✓ Fixed Frame        OK
Grid                   ✓
RobotModel             ✓
PointCloud2            ✓
PointCloud2            ✓
Image                  ✓
TF                     ✓
  Status: Warn
  Show Names           ✓
  Show Axes            ✓
  Show Arrows          ✓
  Marker Scale         1
  Marker Alpha         1
  Update Interval      2
  Frame Timeout        30
  Filter (whitelist)

Synchronization: Off   ROS Time: 00179.49   ROS Elapsed: 73.49   Wall Time: 00179.52   Wall Elapsed: 73.46
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.   31 fps
```

**Fig. 6.** ORB-SLAM with Pepper

## 3.2 Navigation

Navigation is a key element enabling the robot to reach desired locations. There are three sections: mapping, localization, and path planning. The Mapping function generates grid maps by processing laser range data from the GMapping library with very efficient Rao-Blackwellized particles[3]. AMCL (Adaptive Monte Carlo localization) is a probabilistic localization process for two-dimensional robot localization that uses a particle filter to compare a robot's posture to a preset map[11]. Path planning is a robot's ability to discover the optimum path to a destination while avoiding obstacles utilizing the DWA (Dynamic Window Approach)[2], which is often utilized by local planners. The example of our robot navigation is shown in Figure 7.

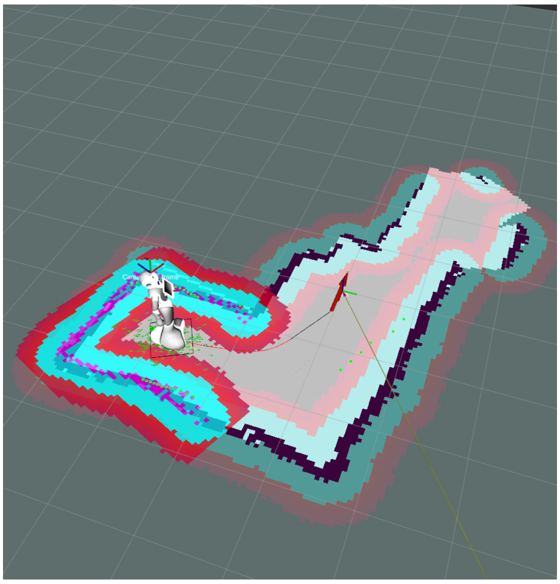


**Fig. 7.** ROS navigation stack with Pepper

## 4 Speech Recognition

In pursuit of robust speech recognition capabilities, the Whisper[9] for Speech-To-Text system was selected due to its proficiency in recognizing multiple languages with superior accuracy compared to Pepper's native speech-to-text SDK. Additionally, for Text-To-Speech functionality, we employ the Pepper Text-To-Speech SDK, chosen for its adequacy and optimization specifically tailored to the acoustic properties of Pepper's speaker system.

## 5 Mission States Planner

SMACH [5] is employed for constructing hierarchical state machines, wherein nodes represent distinct states of execution, and edges delineate the transitions between nodes corresponding to their respective outcomes. The utilization of SMACH facilitates the rapid development of resilient robot behavior characterized by maintainable and modular code.

## 6 Experiment and result

At the qualified social standard platform league category 2024 round the task chosen was Carry my luggage, based on the competition task according to the rules of 2023. The objective of this task focuses on the efficiency of vision by obtaining hand gestures, person detection, human following, and navigation.

*Experiment VDO:* `https://youtu.be/7VMHZ97ns4I`

## 7 Competition result

Our achievements in past competitions in @HOME SSPL, OPL, and EDUCATION league are shown in Table. 1

**Table 1.** Competition Result

| Competition | Result |
|---|---|
| RoboCup 2015 Hefei | @HOME OPL 7th |
| RoboCup Japan Open 2017 Nagoya | @HOME EDU 2nd |
| RoboCup APAC 2017 Bangkok | @HOME OPL 2nd |
| RoboCup 2018 Montreal | @HOME EDU 2nd |
| RoboCup 2019 Sydney | @HOME EDU 1st |
| RoboCup Japan Open 2019 Nagaoka | @HOME EDU 1st |
| RoboCup 2021 Online Challenge | @HOME EDU<br>- Best Technical Paper<br>- People's Choice Award |
| RoboCup APAC 2021 Aichi | @HOME OPL 3rd |
| RoboCup 2022 Bangkok | @HOME OPL 4th |
| RoboCup 2023 Bordeaux | @HOME SSPL 4th |

## 8 Conclusion

In summary, this team description paper outlines our strategy for addressing the challenges posed by RoboCup 2024. The results presented highlight the successful integration of diverse capabilities to tackle the challenge effectively, with ongoing developments to enhance our repertoire. Drawing upon our extensive past experiences, the SKUBA team asserts confidence in our capabilities and qualifications, positioning us as well-prepared contenders for participation in the competition.

## References

1. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)
2. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robotics Automation Magazine **4**(1), 23–33 (1997). https://doi.org/10.1109/100.580977
3. Grisetti, G., Stachniss, C., Burgard, W.: Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. pp. 2432–2437 (2005). https://doi.org/10.1109/ROBOT.2005.1570477
4. Hinterstoisser, S., Pauly, O., Heibel, H., Martina, M., Bokeloh, M.: An annotation saved is an annotation earned: Using fully synthetic training for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
5. Hudson, N., Ma, J., Hebert, P., Jain, A., Bajracharya, M., Allen, T., Sharan, R., Horowitz, M., Kuo, C., Howard, T., et al.: Model-based autonomous system for performing dexterous, human-level manipulation tasks. Autonomous Robots **36**, 31–49 (2014)
6. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J., et al.: Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172 (2019)
7. Moore, T., Stouch, D.: A generalized extended kalman filter implementation for the robot operating system. In: Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13). Springer (July 2014)
8. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103
9. Radford, A., Kim, J.W., Xu, T., Brockman, G., Mcleavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 28492–28518. PMLR (23–29 Jul 2023), `https://proceedings.mlr.press/v202/radford23a.html`
10. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7464–7475 (June 2023)
11. Zhang, B., Liu, J., Chen, H.: Amcl based map fusion for multi-robot slam with heterogenous sensors. In: 2013 IEEE International Conference on Information and Automation (ICIA). pp. 822–827 (2013). https://doi.org/10.1109/ICInfA.2013.6720407

## Pepper Software and External Devices

We use a standard SoftBank Robotics Pepper robot unit.

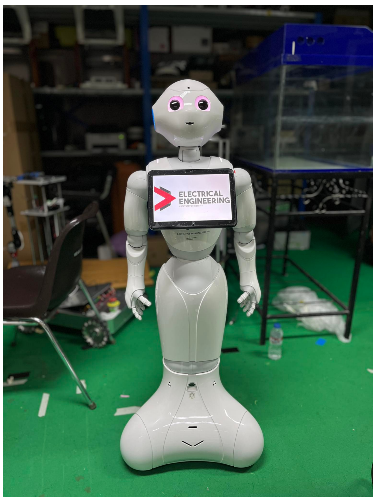


**Fig. 8.** Pepper Robot

## Robot's Software Description

*For our Pepper robot, we are using the following software:*

- Platform: Ubuntu 16.04 (Xenial Xerus) / Ubuntu 20.04 (Focal Fossa)
- ROS version: Kinetic/Noetic
- Face recognition: Naoqi APIs
- Object recognition: Yolo-tiny V7
- Speech interaction: Whisper & Naoqi APIs
- Pose estimation: OpenPose & MediaPipe
- Manipulation: Moveit!

## External Devices

*Pepper robot relies on the following external hardware:*

### Main Computer

- CPUs: Intel core i7-7700HQ
- GPUs: Nvidia GTX 1070
- RAM: 16GB
- OS: Ubuntu 20.04

### Server for Naoqi SDK

- Virtual Machine using VMWare
- CPUs: 4vCPU
- RAM: 8GB
- OS: Ubuntu 16.04