

RFC-Tsudanuma - Extended Abstract

Humanoid Soccer League RoboCup 2026 - Incheon, South Korea

Satoshi Inoue, Ikuo Shige, Yuto Aiba, Haruki Ogawa, Yuugo Nishio, Masafumi Horiguchi, Suzuha Kiuchi, Keita Takakura

Chiba, Japan <https://github.com/RFC-Tsudanuma>
rfc.tsudanuma@gmail.com

Abstract. This paper introduces the team and technologies of RFC-Tsudanuma, participating in the RoboCup World Championship for the first time. We aim to achieve stable walking and team play based on reinforcement learning. By combining these with robust self-localization, including visual odometry, we aim to achieve higher-quality soccer performance.

Keywords: humanoid robotics · bipedal robot · reinforcement learning · robot soccer · RoboCup Soccer.

1 Introduction

We are an independent team of Japanese students and professionals. We do not represent any specific university or company. Many of our members have research experience in robotics and humanoid robotics at university. We launched the team in the summer of 2025 and have participated in the RCAP Beijing Masters and RCAP Abu Dhabi. In those previous two competitions, we competed using borrowed Booster-T1[1] robots. For this competition, however, we have acquired our own robots and will participate with them for the first time.

2 Robot system

We use two types of robots. The first is the Booster-K1 RoboCup Edition[2] manufactured by Booster Robotics. We use this robot without modification. The second is the Mini-Pi[3] manufactured by High torque Robotics. Since this robot consists only of a lower body, we designed and integrated a custom upper body.

We constructed the ROS2 based software architecture with a loosely coupled design. We present a schematic overview of the software system in the following Fig.1.

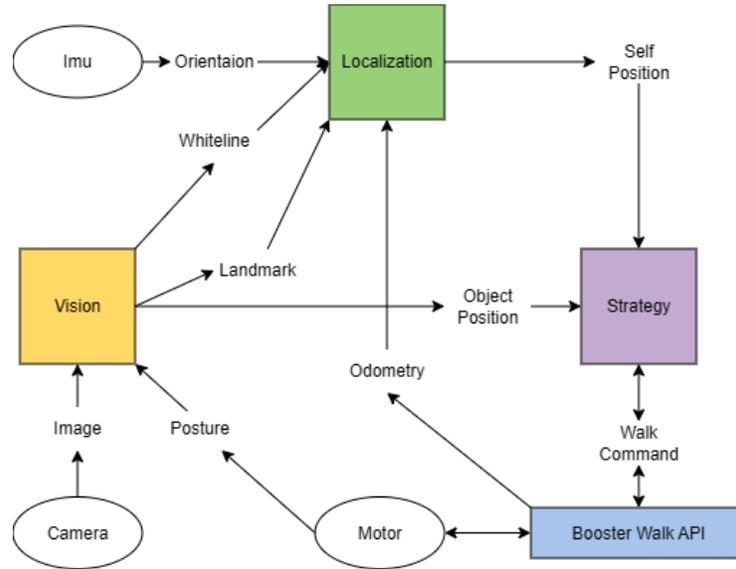


Fig. 1. Software system overview

3 Locomotion

In our previous competitions, we utilized the built-in walking policy of the Booster-T1. However, through these experiences, we identified several issues. First, the inaccurate odometry adversely affected self-localization. Furthermore, tuning the walking motion was difficult and we could not effectively control the direction of the ball during kicks. To address these issues, we implement a custom walking policy based on reinforcement learning. We plan to train two gaits: one for normal walking that prioritizes odometry accuracy and another that enables a seamless approach to the ball and achieving the walk kick.

4 Strategy

We currently utilize a Hierarchical task network planner[4] for decision-making. While this method facilitates the tuning of individual actions, such as implementing specific penalty plays, it requires us to manually write a vast number of conditional branches to implement cooperative plays like passing. This process is inefficient and difficult. To resolve this issue, we plan to incorporate the potential field method into part of our decision-making logic. Determines the spaces for passing and coordinated positioning by generating a potential field using objects on the field as elements.

5 Perception and Localization

RoboCup Soccer environments are characterized by frequent robot interactions, partial occlusions, and disturbances caused by falls and collisions. To address these conditions, we construct a perception and localization system that combines real-time visual recognition with particle filter-based localization.

5.1 Object Recognition and Segmentation

Object detection is performed using YOLOv8[5]. While Transformer-based detectors such as RT-DETR[6, 7] have shown strong performance in terms of detection accuracy, YOLOv8 is adopted in this work due to its lower inference latency and stable performance on edge computing hardware. This allows visual observations to be provided to the localization module at a sufficient rate for real-time state estimation.

White field lines are detected using a semantic segmentation approach based on Segmentation Models PyTorch (SMP)[8]. A U-Net architecture with a ResNet34 backbone is employed as a balance between model capacity and computational cost. To reduce sensitivity to variations in lighting conditions and image noise, data augmentation is applied during training using Albumentations.

5.2 Localization with EMCL

For self-localization, we use Expansion Resetting Monte Carlo Localization (EMCL)[9]. EMCL extends standard Monte Carlo Localization by introducing particle expansion and resetting mechanisms, which are intended to recover localization when the robot’s belief becomes inconsistent with sensor observations, such as after falls or external disturbances.

Monte Carlo Localization Process The localization process follows a standard particle filter cycle:

- **Prediction:** Each particle is propagated using odometry information. Motion noise is added in proportion to the estimated velocity to account for uncertainty in walking motion.
- **Update:** Particle weights are computed using visual observations. Landmark detections obtained from YOLOv8 and white line observations from the segmentation model are both incorporated into the likelihood calculation.
- **Resampling:** Particles are resampled based on their weights using the effective sample size criterion. When the particle distribution becomes overly concentrated or inconsistent, EMCL introduces additional particles to maintain hypothesis diversity.

White Line-Based Likelihood Estimation To evaluate the consistency between observed white lines and the known field geometry, a pre-defined field map is used. The map is smoothed with a Gaussian filter to tolerate small observation and projection errors. Points extracted from the segmentation output are projected onto the field map, and a raw matching score is computed.

This raw score is converted into a likelihood value using a sigmoid function:

$$L = \frac{1}{1 + \exp(-20 \times (raw_score - 0.2))}$$

This transformation limits the influence of extreme scores and reduces sensitivity to segmentation noise and partial mismatches.

5.3 System Integration

The perception and localization modules are integrated such that visual observations are continuously supplied to the EMCL framework. By combining object detections and white line observations, the system maintains a consistent pose estimate during normal operation and can recover localization when significant disturbances occur. This design is intended to provide reliable localization under typical RoboCup match conditions, including physical contact and unstable walking.

References

1. Booster-Robotics Booster-T1 <https://www.booster.tech/booster-t1/>,2026
2. Booster-Robotics Booster-K1 <https://www.booster.tech/booster-k1/>,2026
3. High Torque Robotics Mini-Pi. <https://www.hightorquerobotics.com/pi/>,2026
4. Erol, Kutluhan and Hendler, James and Nau, Dana S. HTN planning: complexity and expressivity. *AAAI*, pp. 1123–1128, 1994.
5. Glenn Jocher et al. Yolov8. <https://github.com/ultralytics/ultralytics>, 2023.
6. W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu. RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer, 2024.
7. Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen. DETRs Beat YOLOs on Real-time Object Detection, 2023.
8. P. Iakubovskii, Segmentation Models PyTorch, GitHub repository, 2019. Available at: https://github.com/qubvel/segmentation_models.pytorch.
9. Ryuichi Ueda, et al. Expansion Resetting for Recovery from Fatal Error in Monte Carlo Localization – Comparison with Sensor Resetting Methods. *Proc. of IROS*, pp. 2481–2486, 2004.