Fukuro Team Description Paper 2026: Innovations in Hybrid Localization, Unified Segmentation, and Robust Electro-Mechanical Architecture

Pradana Yahya Abdillah, Salis Haidar Luthfi, Ari Ahmad Wiyoga, Ardra Aghfarrel Setiyono, Mohammad Lathif Adani, and Maulana Isqi Habibah

Universitas Gadjah Mada
C20 GMRT Research Center, Yogyakarta, Indonesia
fukuro.ugm@gmail.com
https://fukurougm.github.io

Abstract. Team Fukuro is an autonomous robotics team preparing for its debut at the RoboCup Middle Size League (MSL) 2026. This Team Description Paper presents our foundational developments in hardware and software as we step into international competition. Recognizing the highly dynamic and unpredictable nature of the MSL environment, our research focuses on maximizing computational efficiency, hardware safety, and system robustness. We present a highly isolated electromechanical architecture utilizing Zero Voltage Switching (ZVS) and Direct Memory Access (DMA) for deterministic control. In software, our primary scientific contributions include a novel Hybrid GA-AMCL (Genetic Algorithm - Monte Carlo Localization) system providing extreme robustness against severe wheel slip, and a Unified Dual-Layer Vision Architecture utilizing YOLO26 Instance Segmentation. This vision framework introduces an 'Implicit Obstacle Handling' mechanism for optimal, linear-time $O(H \times W)$ target estimation. This paper outlines Fukuro's major milestones toward achieving full autonomy in robotic soccer.

Keywords: RoboCup MSL · Autonomous Soccer Robot · Hybrid GA-MCL · YOLO-Seg · Implicit Obstacle Handling · Distributed Power Isolation.

1 Introduction

The RoboCup Middle Size League (MSL) demands robotic platforms capable of perceiving highly dynamic environments, executing rapid mechanical actuation, and making decentralized tactical decisions. Traditional systems often face bottlenecks in computational latency, sensor drift (e.g., severe wheel slip), and electrical interference between high-power actuators and logic circuits.

As a debuting team, Team Fukuro addresses these challenges from the ground up. This paper elaborates on our holistic robot design. Section 2 details our mechanical actuation and distributed electrical safety. Section 3 outlines the modular software framework. Sections 4 and 5 present our core scientific contributions: an evolutionary Hybrid GA-AMCL localization engine and a unified deep-learning vision system capable of implicit obstacle avoidance.

2 Electro-Mechanical Architecture

2.1 Dynamic Actuation and Ball Handling

Our mechanical structure is engineered to maximize agility, traction, and ball retention under aggressive maneuvers. The primary kicking mechanism utilizes a linear solenoid actuator operating on electromagnetic induction to deliver impulsive force to the kicker leg. To enable dynamic trajectory adjustments—essential for chipping or lob passing—the robot is equipped with a pulley-driven elevation system. This mechanism employs a PG28 motor coupled with a high-tensile fishing line. Modulating the motor's rotation winds or unwinds the line, providing precise vertical displacement of the kicker's impact point.

For ball handling, the system relies on twin PG45 motors coupled with rubber-rimmed wheels to achieve maximal surface traction. To mitigate high-frequency vibrations and maintain constant contact with the ball, the mechanism integrates an active upper dampening system. Furthermore, bottom auxiliary wheels are mounted to stabilize the ball's position, ensuring it remains strictly within the robot's control axis even during rapid lateral movements.

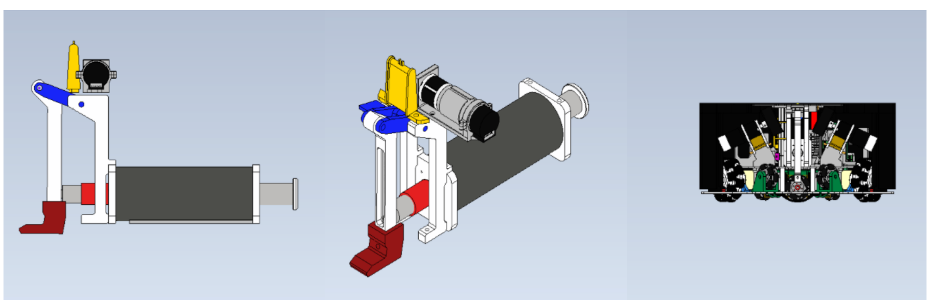

Fig. 1. 3D CAD Render of the Fukuro kicking mechanism and ball handling system, highlighting the PG45 motors and pulley-driven elevation.

2.2 Distributed Power Isolation and Safety

The electronic architecture is built upon the core principles of distributed computing and power isolation. To prevent transient voltage droop during aggressive maneuvers, the power distribution is segregated into four isolated domains utilizing a 6S3P Lithium-based dual-battery topology:

- **19V DC:** Dedicated to the High-Level Controller (Intel NUC12WSHi5) for AI inference.
- **20V DC:** High-current rail for PG36/PG45 motor drivers, protected by a 20A fuse to handle massive inrush currents.
- **>200V DC:** An independent high-voltage subsystem for the solenoid kicker.
- **5V/3.3V DC:** Regulated lines for the MCU logic (STM32 Blackpill/Bluepill), protected by a dedicated 10A fuse.

The high-voltage kicker subsystem utilizes a Zero Voltage Switching (ZVS) oscillator to efficiently step up voltage. To safely handle the massive transient pulse required to discharge the capacitors, we employ an Insulated-Gate Bipolar Transistor (IGBT) due to its superior durability. Crucially, a ceramic cement bleeder resistor is permanently wired across the capacitor bank. This automated safety interlock ensures the high-voltage capacitors are safely discharged when powered off, adhering to strict RoboCup safety standards.

2.3 Low-Level Control and Signal Integrity

Low-level operations are governed by STM32 microcontrollers executing an Interrupt-Driven Firmware Architecture coupled with Direct Memory Access (DMA). This offloads the core CPU from I2C (IMU data) and UART (NUC communication) handling, ensuring zero-latency kinematics calculations. Furthermore, all custom PCBs are designed in-house adhering to IPC-2221 standards. High-current tracks to the BTS7960 H-bridges are reinforced with applied solder tinning to reduce parasitic resistance, while decoupling capacitors are placed adjacently to all MCU power pins to filter Electromagnetic Interference (EMI).

To seamlessly bridge this low-level hardware execution with the high-level AI processing, a robust communication topology is implemented. The High-Level Controller (Intel NUC) interfaces with the Master MCU via a reliable USB connection using an FTDI serial converter, selected specifically for its driver stability under Linux/ROS to ensure low-latency transmission of velocity vectors [1]. Concurrently, an internal UART bus facilitates inter-MCU task distribution between the primary kinematics controller and the secondary I/O management MCU [1]. This strict hardware communication decoupling forms a solid foundation for our high-level software framework.

3 Modular Software Architecture

The Fukuro software stack is partitioned into distinct functional layers communicating via TCP/Protobuf to ensure system scalability.

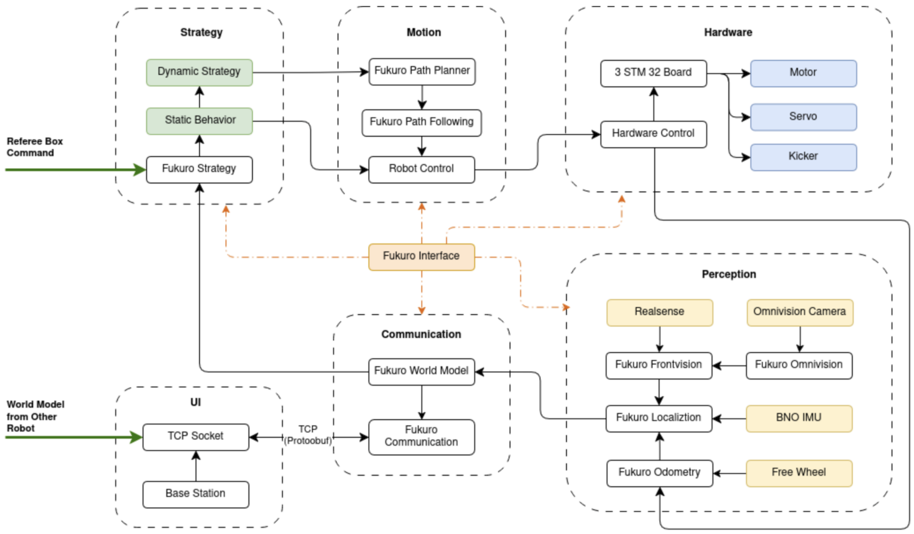

Fig. 2. The Fukuro Modular Software Architecture detailing the TCP/Protobuf integration between Strategy, Perception, Motion, and Hardware layers.

The *Perception Layer* continuously fuses data from the RealSense camera, Omnivision sensor, IMU, and free-wheel odometry to construct the Local World Model. The *Strategy Layer* evaluates tactical decisions and generates specific motion targets, forwarding them to the *Motion Layer* (Path Planner and Robot Control). Physical execution is handled by the *Hardware Layer* through the STM32 boards, ensuring a strict decoupling between high-level AI processing and hardware-level actuation.

4 Hybrid GA-AMCL Self-Localization System

4.1 Motivation and Omnidirectional Perception

Conventional Adaptive Monte Carlo Localization (AMCL) heavily relies on wheel odometry, leading to severe drift when slipping occurs. Inspired by evolutionary localization advancements in robot soccer [5], Team Fukuro introduces a novel Hybrid GA-AMCL Architecture. This system fuses the rapid, gradient-free optimization of a Genetic Algorithm (GA) with the probabilistic motion prediction of AMCL [1].

The perception layer utilizes a 360° hyperbolic mirror omnivision camera. To handle uneven stadium lighting, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied before extracting white field lines via HSV thresholding [6]. An optimized perspective transformation converts the polar image into a bird's-eye view using a pre-computed 3rd-order polynomial Look-Up Table (LUT) ($r_{pixel} = a_3d^3 + a_2d^2 + a_1d + a_0$), achieving sub-millisecond execution.

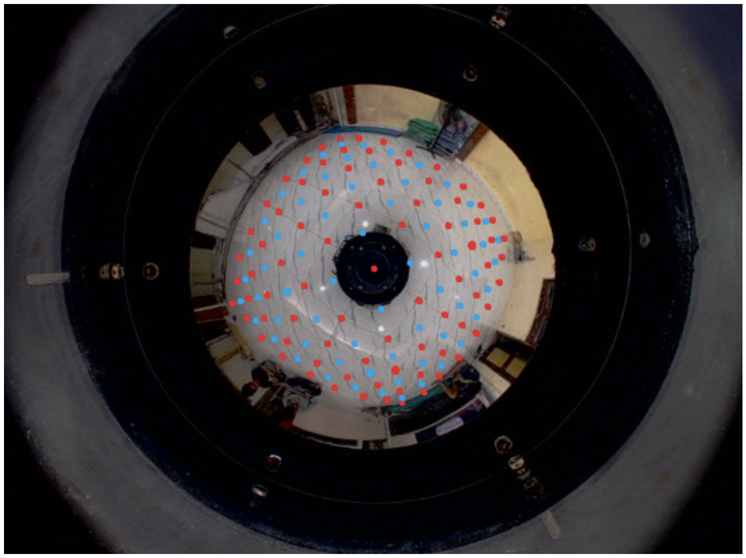

Fig. 3. Omnidirectional camera calibration mapping real-world distances to pixel coordinates using a 3rd-order polynomial LUT approach.

A Zhang-Suen skeletonization process [7] then extracts a "Search Model" comprising up to 200 spatial points.

4.2 The Hybrid GA-AMCL Engine

The robot's global pose (x, y, θ) is encoded as a 29-bit chromosome using Reflected Gray Code (yielding ∼0.6–0.8 cm and ∼0.7° resolution). Gray code guarantees that a single bit-flip mutation results in a proportional spatial shift, mimicking AMCL's controlled Gaussian noise but within a highly structured discrete space [4]. The estimation loop follows a strict sequence:

1. **AMCL Motion Prediction (Warm Start):** Before evolution, the inherited population is shifted using odometry data $\Delta\mathbf{q}_{odom}$ combined with a probabilistic noise model.
2. **Fitness Evaluation:** Each candidate is evaluated based on the overlap between the visual Search Model and a dilated field map.
3. **Evolutionary Recombination (The GA Advantage):** Unlike AMCL which only resamples, our system utilizes a *One-Point Crossover*. This actively recombines partial solutions (e.g., extracting an accurate (x, y) from Parent A, and θ from Parent B). Combined with a mutation rate of $p_m = 0.05$, this accelerates convergence to just 3-5 generations at 30 FPS, providing extreme robustness against high-speed movements [3].

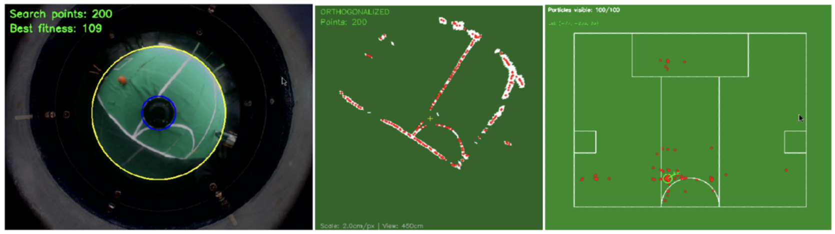

Fig. 4. Hybrid GA-AMCL in action. (Bottom Left) Omnidirectional input. (Bottom Right) Orthogonalized bird's-eye view. (Top) Field map showing the estimated pose (green circle) and concentrated GA population particles (red dots).

4.3 Kidnapped Robot Recovery

To handle forced displacements (e.g., by the referee), the system monitors the peak fitness score. If the fitness drops below a threshold ($\tau = 15.0$) for 10 consecutive frames, an AMCL-validated recovery triggers: 50% of the lowest-fitness population is randomly reinitialized across the field (global exploration), while the top 50% is retained (local exploitation) as a fail-safe against temporary visual occlusions.

5 Unified Vision System and Object Segmentation

5.1 Transition to Multi-Class Instance Segmentation

Deploying multiple algorithms for balls, opponents, and goals creates severe computational bottlenecks. Extending concepts from fully neural object detection paradigms in robot soccer [8], Team Fukuro developed a Unified Multi-Class Instance Segmentation pipeline utilizing the YOLO26-Seg architecture [10].

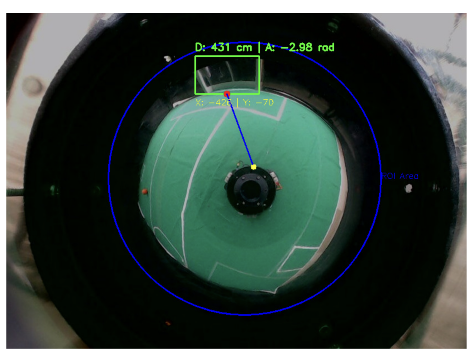

Fig. 5. Omnidirectional global perception continuously estimating the relative distance (D) and angle (A) of the goal area to initiate body alignment.

Operating on a dual-layer camera setup, the omnidirectional camera provides global spatial awareness (Local World Map) to guide orientation (Figure 5). Once aligned, the front-facing Intel RealSense D455 depth camera delivers high-fidelity positional tracking for tactical execution, mirroring advanced YOLO-based goal alignment strategies [9].

5.2 Implicit Obstacle Handling and Target Estimation

When executing a shot, the line of sight is frequently occluded by opponents. Instead of computing explicit geometric bounding-box collisions via multiple models, our system leverages **Implicit Obstacle Handling**. When an opponent stands in front of the goal, the YOLO26-Seg model naturally excludes the occluded pixels, resulting in a topological "hole" within the goal's mask.

Fig. 6. Implicit Obstacle Handling. (Left) The Black obstacle naturally breaks the segmentation mask. (Right) Clear view of the goal. The algorithm dynamically discovers the largest open area (green rectangle) to establish a safe ground-level target point (red dot).

To extract the optimal shooting trajectory, we execute a deterministic $O(H \times W)$ target estimation algorithm (Figure 6):

1. **Morphological Erosion:** The mask is eroded to create a safety margin from goalpost edges and obstacles.
2. **Largest Inscribed Rectangle Search:** A histogram-based stack sweep dynamically isolates the largest contiguous, unobstructed shooting area.
3. **Vertical Projection:** The horizontal center of this rectangle ($target\_x$) is computed and projected vertically down to the lowest active pixel ($target\_y$), ensuring the target corresponds to a valid ground-level entry point.

Finally, the precision aiming angle (θ) is computed as $\theta = \text{atan2}(target\_x - robot\_x, robot\_y - target\_y)$, directly providing obstacle-free shooting trajectories to the motion controller.

6 Conclusion

This paper outlined Fukuro's holistic approach to the RoboCup MSL. By integrating a distributed electronic architecture with custom ZVS high-voltage actuation, a Hybrid GA-AMCL localization engine, and a linear-time $O(H \times W)$ Implicit Obstacle Handling segmentation pipeline, we have drastically reduced computational bottlenecks while ensuring extreme robustness. These milestones solidify Team Fukuro's readiness for our international debut in 2026.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
2. Watanabe, K., Ma, Y., Suzuki, H.: Real-time Self-localization using Model-based Matching for Autonomous Robot of RoboCup MSL. J. Robot. Netw. Artif. Life **7**(2), 98–102 (2020). `https://doi.org/10.2991/jrnal.k.200528.005`
3. Ma, Y., Watanabe, K., Kono, H., Suzuki, H.: Verification of Robustness against Noise and Moving Speed in Self-localization Method for Soccer Robot. J. Robot. Netw. Artif. Life **8**(1), 66–71 (2021). `https://doi.org/10.2991/jrnal.k.210521.015`
4. Watanabe, K., Ma, Y., Suzuki, H.: A Self-Localization Method Using a Genetic Algorithm Considered for Environmental Change. J. Adv. Comput. Intell. Intell. Inform. **26**(1), 32–40 (2022). `https://doi.org/10.20965/jaciii.2022.p0032`
5. Wu, C.-H., et al.: An Evolutionary Method to Vision-based Self-localization for Soccer Robots. Scientia Iranica **23**(3), 1224–1235 (2016).
6. Lu, H., Yang, S., Zhang, H., Zheng, Z.: A robust omnidirectional vision sensor for soccer robots. *Mechatronics* **21**(2), 373–389 (2011). `https://doi.org/10.1016/j.mechatronics.2010.06.007`
7. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Communications of the ACM **27**(3), 236–239 (1984)
8. Szemenyei, M., Estivill-Castro, V.: ROBO: Robust, Fully Neural Object Detection for Robot Soccer. In: RoboCup 2019: Robot World Cup XXIII. LNCS, vol. 11531. Springer, Cham (2020).
9. Bahagia, D., et al.: Enhancing Humanoid Robot Soccer Ball Tracking, Goal Alignment, and Robot Avoidance Tasks Using YOLO-NAS. J. Robot. Control **4**(3) (2023).
10. Wang, A., Liu, B., Zhang, Y., et al.: YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection. arXiv preprint arXiv:2509.25164 (2025).