Software Survey 2026

Team name

Blenders FC

Which division(s) are you applying for? If your used software differs between divisions, please fill out the survey once per division.

Small Size (height < 110 cm, weight < 15 kg)

Is your software fully or partially OpenSource? If so, where can it be found?

https://github.com/Blenders-FC

Are you using any software developed by other teams? If so, list every component that you are reusing and the team that originally developed it.

Vision model from Rhoban

Are you using any datasets in your research? If you are using your own datasets, are they public?

No

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).

N/A

Are there any other contributions you would like to share with the RoboCup community?

Fix OP3 repos for Ubuntu 22.04

Which approach are you using to generate the robot walking motion?

The robot's walking motion is generated using Kajita’s mathematical model, which models the robot as an inverted pendulum under constraint. This ensures balance by keeping the robot’s center of mass within the support polygon, allowing for stable walking motion.

Which approach are you using to generate other motions of the robot (e.g. kicking, standing up)?

Inverse kinematics is used to define a starting and a final position, these are defined in a function that is called when the IMU surpasses a threshold and in between points are interpolated to create a full standing up motion.

To generate kick motions we are using the kinematic model by processing key poses in the movement

Do you have a kinematic or dynamic model of your robot? If so, how did you create it (e.g. measure physical robot, export from CAD model)?

Yes, we have a kinematic model that was developed with official measurements of the physical robot and represented as links where the final effector is the foot in order for it to be always parallel to the floor.

What approaches are you using in your robot's visual perception?

We use a YOLOv8 computer vision model from Rhoban, That enables the robot to detect soccer balls and goal posts

Are you planning with objects in Cartesian or image space? If you are using Cartesian space, how do you transform between the image space and cartesian space?

We plan in Cartesian space. The transformation from image space to Cartesian space is achieved using camera calibration parameters and trigonometric functions. First, the ball’s pixel coordinates are obtained from the image. Then, using the camera’s intrinsic properties and the known height and tilt angles of the camera, we estimate the real-world (X, Y, Z) position of the ball relative to the robot

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?

Yes, our robots use a computer vision model to detect objects within its camera frame. When specific objects are identified (such as balls or goal posts), the robot dynamically adjusts its camera to gather more information or maintain focus on the detected objects. This will enable the robot to perform actions such as preparing for kicking the ball.

What approach are you using to localize your robot?

Our robot is currently undergoing localization testing using different algorithms. We are evaluating the performance of ORB-SLAM and VINS-Mono to determine the most suitable real-time option. Additionally, we are developing a custom localization algorithm that uses goal triangulation and field line positions to estimate the robot's position within the soccer field.

Is your team performing team communication? Which communication protocol are you using?

We have implemented and tested Protobuf communication from the NuBots team, Protobuf message protocol for the Robocup Humanoid league, and we are currently deciding which information from our robots will be transmitted by this protocol.

What approach are you using for navigation? Are you avoiding obstacles?

Currently, the robot does not implement a full path planning algorithm that takes obstacles into account, but future iterations may incorporate a path planning algorithm to navigate while avoiding obstacles, such as potential fields.

How is the behavior of your robots structured? (e.g. Behaviour Trees)

The robot's behavior is structured using Behaviour Trees, where it transitions between different action nodes based on the condition nodes

Are you simulating your robot? If so, which simulator are you using and for what purpose do you use simulations?

We are simulating our robots in Gazebo, mainly with the intention to test localization software in ideal conditions and a full field, since we don't have one. Other secondary uses are for general testing for proofs of concept of robot behavior and algorithms.
We are also simulating the kinematic calculations in MATLAB, to visualize and generate the walking algorithm along with key poses

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?

Ubuntu 22.04 and ROS2 Humble

Is there anything else you would like to share that did not fit any previous question?