Contribution Poster - RoboCup 2018

RobotING@TUHH e.V., Hamburg University of Technology

HULKs
RoboCup SPL Team • TU Hamburg, Germany

Introduction

This document serves as partial fulfillment of the rule book requirements for RoboCup 2018. For this purpose we present a sample of our contributions and research conducted in preparation to RoboCup 2018.

Vision: Horizontal Image Segmentation

Extracting features from the image requires significant data reduction. For this purpose our code featured an image segmenter based on uniformly spaced vertical scan lines. In order to obtain equidistant sampling in world coordinates we added a horizontal image segmenter that utilizes horizontal scan lines. These scan lines are spaced in way that allows approximately constant sampling in world coordinates. For this purpose we use a static projection matrix from a common stand pose.

This new approach allows for better sampling behavior for feature detection. The computational effort over the whole image is reduced while the resolution in the region of distant features can be chosen high enough to detect small features like the black patches on the ball.

Vision: Penalty Spot Detection

In the season of 2017 our vision pipeline only extracted line segments and ball readings from the image. Thus, our self-localization was solely based on single line readings. As lines are a common and non-unique feature of the SPL-field, relocalization capabilities where severely limited.

Therefore, we introduced a new penalty spot detection to our vision pipeline. The penalty spot detection uses the horizontally segmented image to extract candidates from white segments of a reasonable size. For each candidate an according segment in the vertical segmenter image is localized. The intersection of these segments is used to as an initial guess for the penalty spot center. For each of these penalty spot candidates multiple points are sampled from an elliptic environment. The candidate is then classified based on several constraints on chroma, luminance and color of these samples.

Brain: Team Obstacle Model

In the season of 2017 we introduced a team ball model to our code base. As this significantly improved the ball tracking performance over the whole team, we introduced a similar idea for obstacle filtering in 2018.

For this purpose the `TeamObstacleFilter` assembles a global obstacle map from the local obstacle estimates of the individual team members. Obstacles close to each other with a compatible type are merged and a combined position estimate is obtained. This procedure allows to reclassify unknown obstacles by considering information from different sensors and team members. Also, this map is used to add field obstacles (e.g. goal posts) to the obstacle knowledge.

The team obstacle model is used for motion planning and is considered in behavior decisions like kicking and dribbling.

Brain: Localization using Multi-Hypothesis UKF

Until RoboCup 2017 our team used a localization strategy based on a particle filter. In 2018 we introduced a new localization strategy based on an multi-modal Unscented Kalman-Filter (UKF). The filter uses a separate UKF-mode for each localization hypotheses. The hypotheses are rated based on the average feature association error. Hypotheses with sufficient similarity are merged in order to eventually converge to a single hypothesis.

These changes combined with the new feature extractions introduced to the vision pipeline allow for a significant improvement of the localization performance. Additionally, the run-time of the localization module could be reduced by 82%.

Motion: New Kick

The kick motion that was designed in the season of 2016 showed mayor weaknesses in the season of 2017. The artificial turf made it very imprecise and slowed the ball down a lot. Therefore, for this season we designed an new kicking motion. Our framework now features a kick engine that can load generic kicks. Each kick consists of a balancing phase, a kicking phase and a retracting phase. This framework was used to provide two new kicking motions, a strong forward kick and a side kick. Our striker seeks to use these new skills for better tackling behavior.

Motion: Collision Avoidance - Arm Motion

Our motion engine now allows to perform an arm motion to avoid collisions while walking or standing. The motion is designed in a way not to affect the position of the center of mass. Even though walking with a swinging arm motion results in a better gait walking with arms pulled tight to the body is still stable. However, using the arm pull back motion also increases the slip and thus results in worse odometry estimates.

We use this motion in any situation where a collision in predicted by our world model. Collisions are predicted based on obstacles created from team player positions, sonar readings and visual detection.

Debug Tool: Monitor and Test Environment (MATE)

Until 2017 our code featured a web based debug tool written in `node.js`. In this season we developed a new tool using `pyQt5`. This stand-alone tool provides the user with a variety of modular views:

- **text-view** to directly display the serialized data comming from the NAO
- **image-view** to stream debug images rendered on the NAO
- **plot-view** for real-time visualization of numerical data
- **map-view** to render debug-images on the MATE client. E.g. rendering team player knowledge on a spl-field drawing.
- **config-view** to get, set and export configuration data of the NAO.

MATE - Example Debug Layout