# World

**Short description**

World takes the raw vision detection data from multiple cameras which is merged and filtered into a single as perfect as possible world state. Also robot velocities are computed and saved in the state. This world state is outputted on the world\_state topic.

TODO this year: fix the yellow and blue buffer names, they should be us and them and checking for param our\_color is not necessary anymore.

**Executables**

* Filtered\_world
* Dummy\_world: an initial version which directly pipes the raw vision data to the world\_state topic without merging or filtering.

**Dependencies**

* -

**Globals**

* -

**Params**

* -

**Input topics**

* vision\_detection

**Output topics**

* world\_state

**Details**

The callback function of the vision\_detection topic. It keeps track of the number of camera’s active using a map. The incoming raw data of the camera’s is buffered. If all recognized camera’s generated have generated new detection frame, the frames are merged. Camera images can overlap so a robot is seen on multiple camera’s. For the robot the position over multiple robots with the same ID is averaged, for the ball the ball the is closest to the previous ball is chosen. The velocity of the robots and ball is calculated by the predictor class. The predictor also has the possibility to request the position of a specific robot in a given amount of seconds in the future. Extrapolation is used for this.

World could be improved by handling robots and ball when they shortly disappear, maybe by extrapolating the currents robot’s or ball’s position according to its speed.