TU/e

EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

AutoRef Project

Final presentation

MSD Team

Mahmoud Abdelhady
Designer

Aneesh Ashok
System Architect

Ivan Kolodko
Project Manager

Gijs Linskens
Designer

Maryam Mashayekhi
Designer

Lars Maxfield
Team Leader

Anand Vazhayil
Designer

Haoyu Zhu
Designer

Contents

1 Problem description

2 Approach

3 Functional specification

4 Design

5 Conclusions

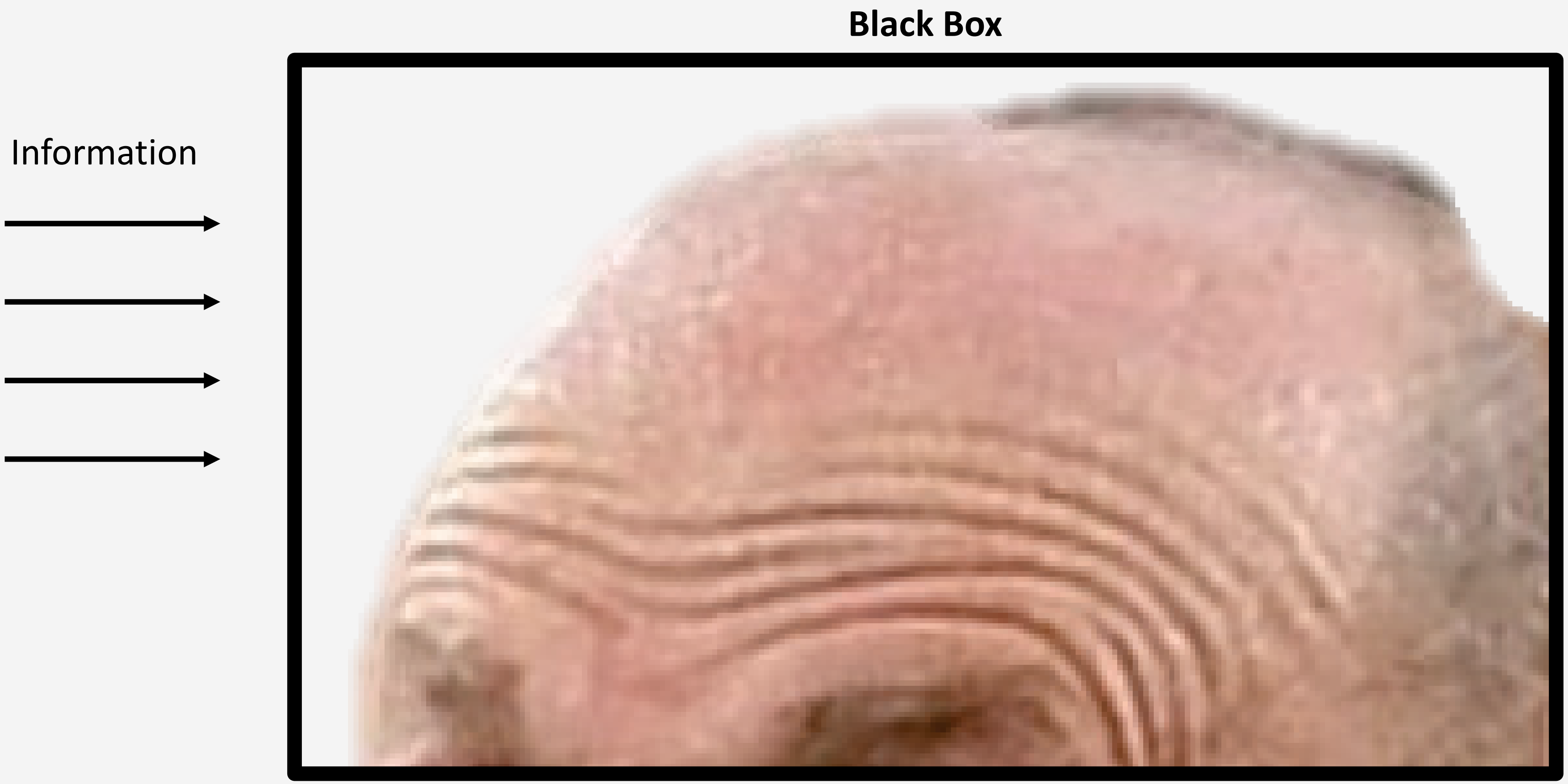
6 Future work

1 Problem description

1

Problem description

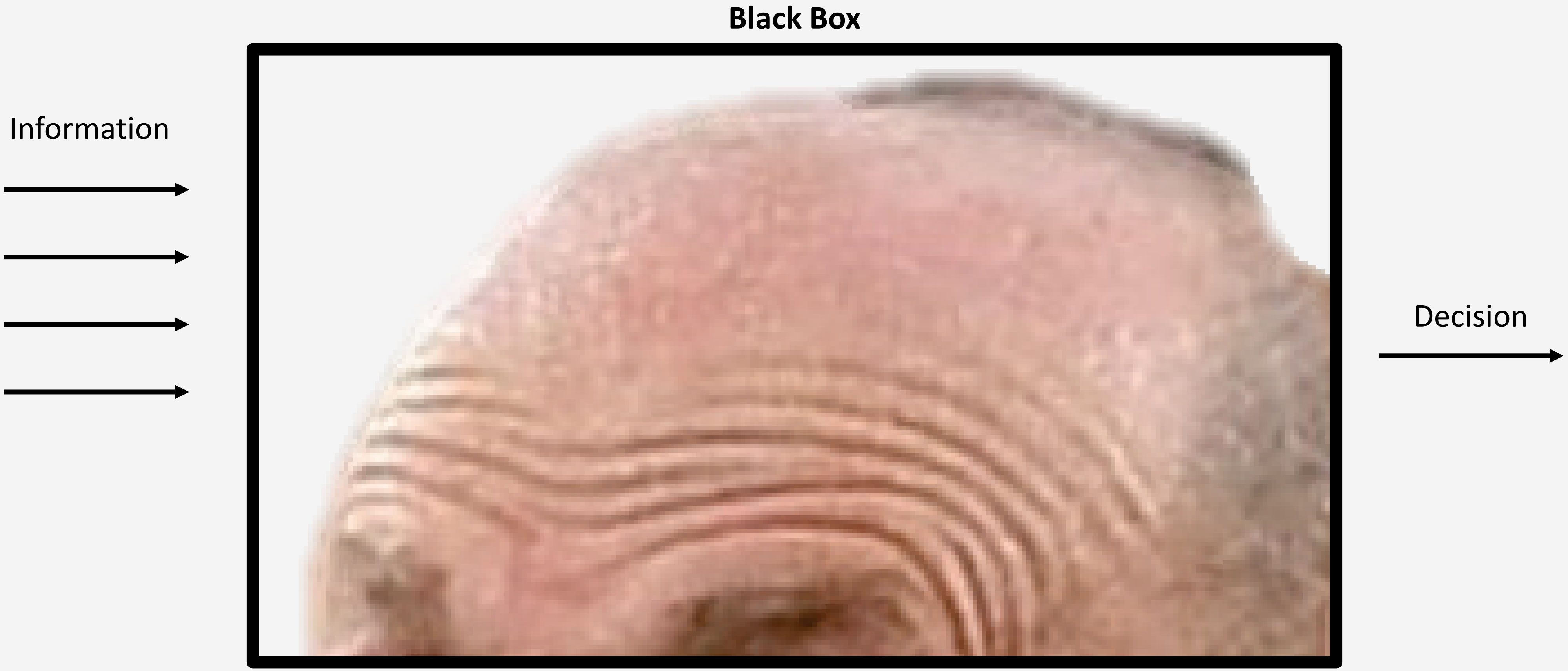
1 Problem description



Black Box

Information

1 Problem description

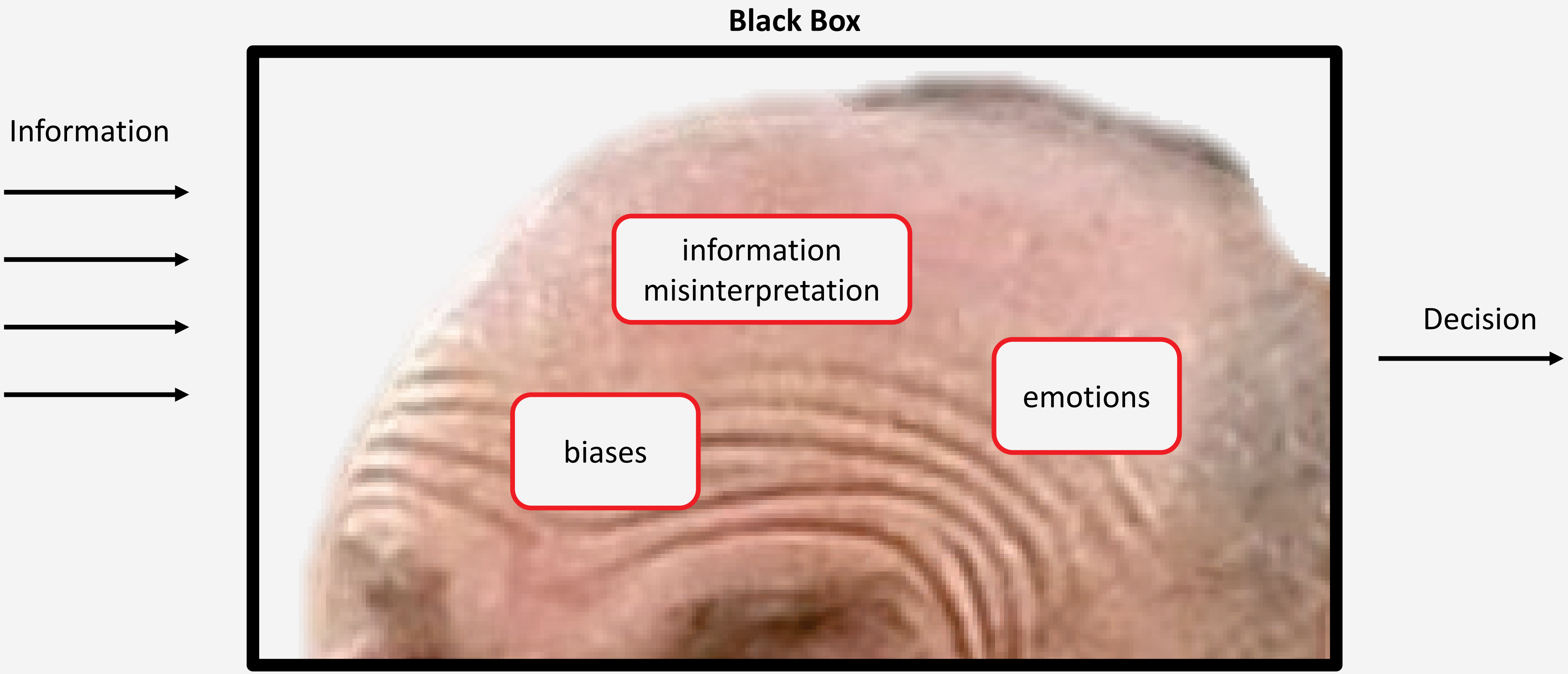


Black Box

Information

Decision

1 Problem description



Black Box

Information

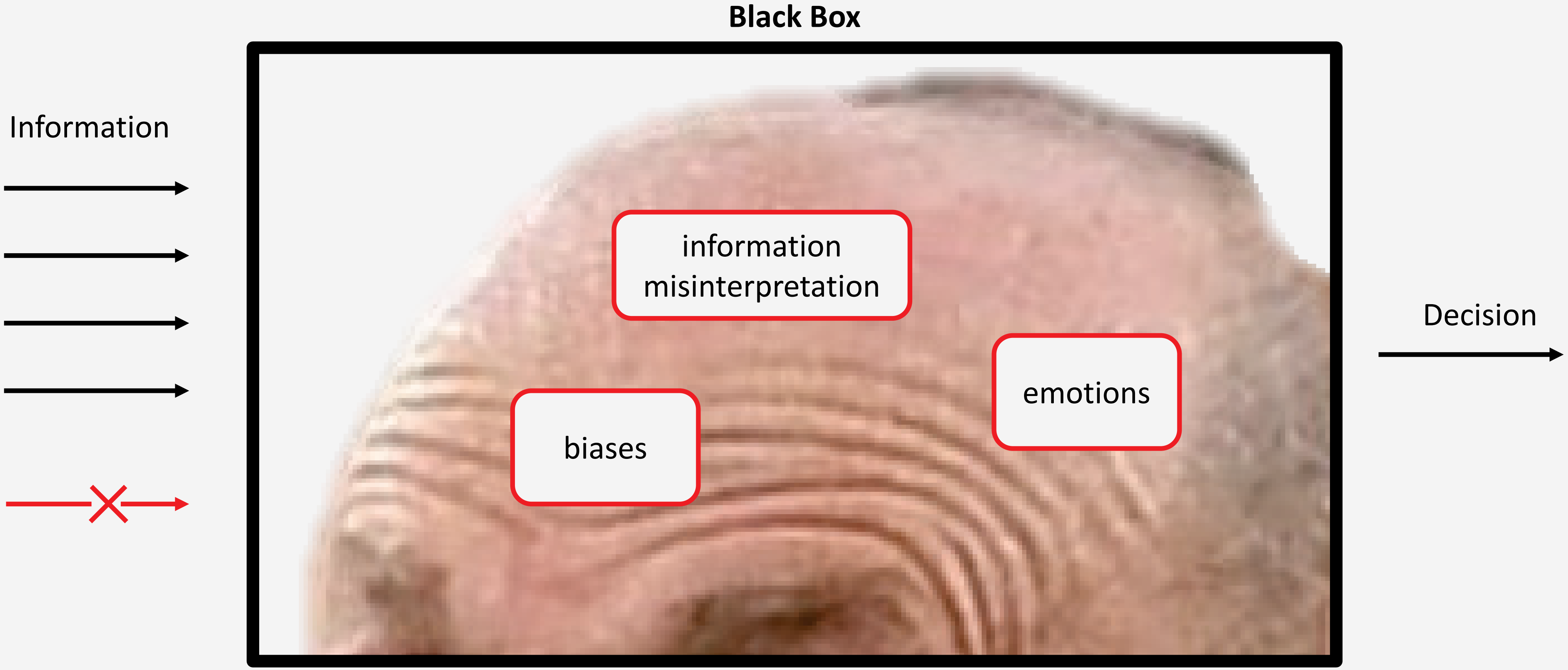
information misinterpretation

biases

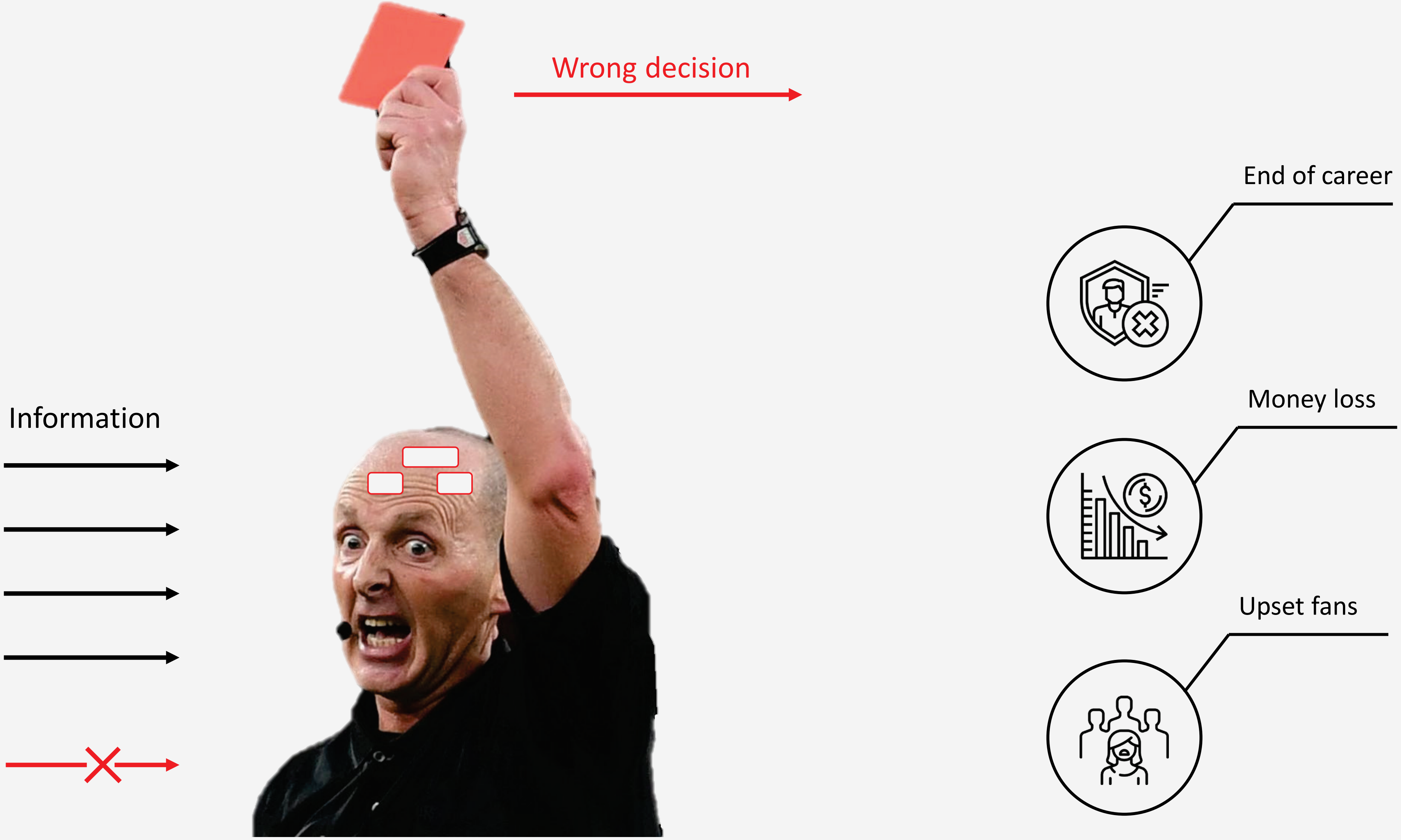
emotions

Decision

Problem description



1 Problem description



Information

Wrong decision

End of career

Money loss

Upset fans

1 Problem description

Project purpose:

Design a system capable to substitute a human-being referee during RoboCup MSL soccer matches

2 – Approach

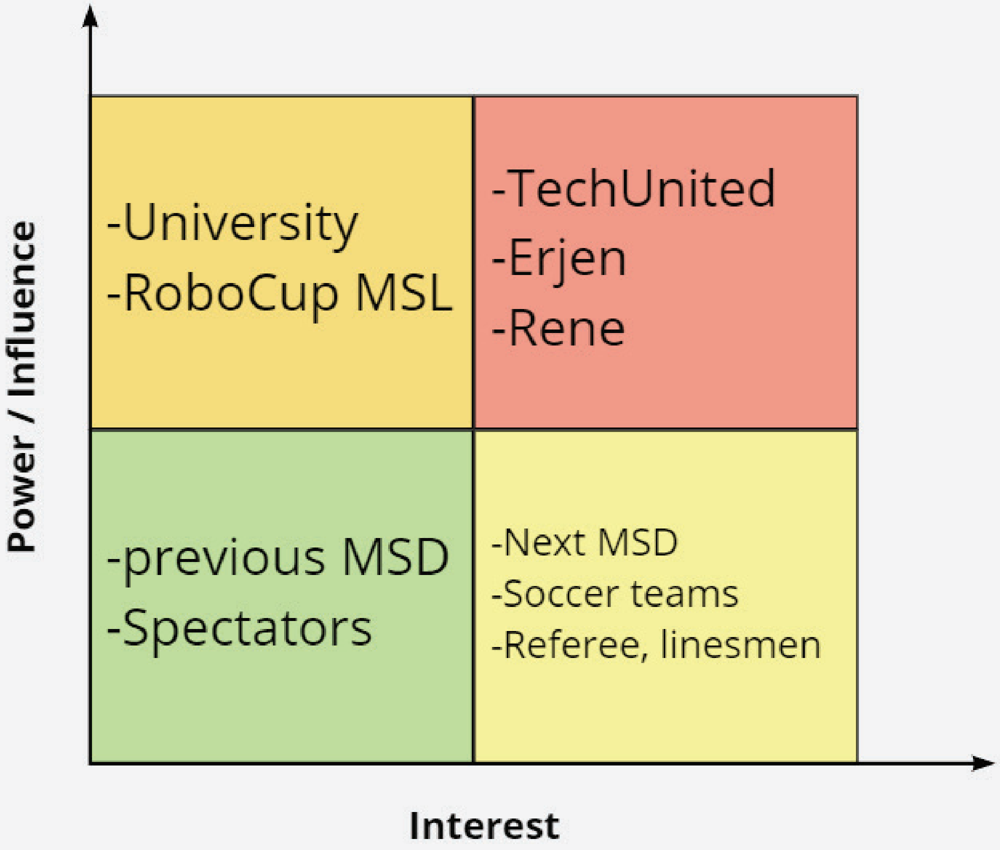
Previous work

5 MSD generations

Major concerns

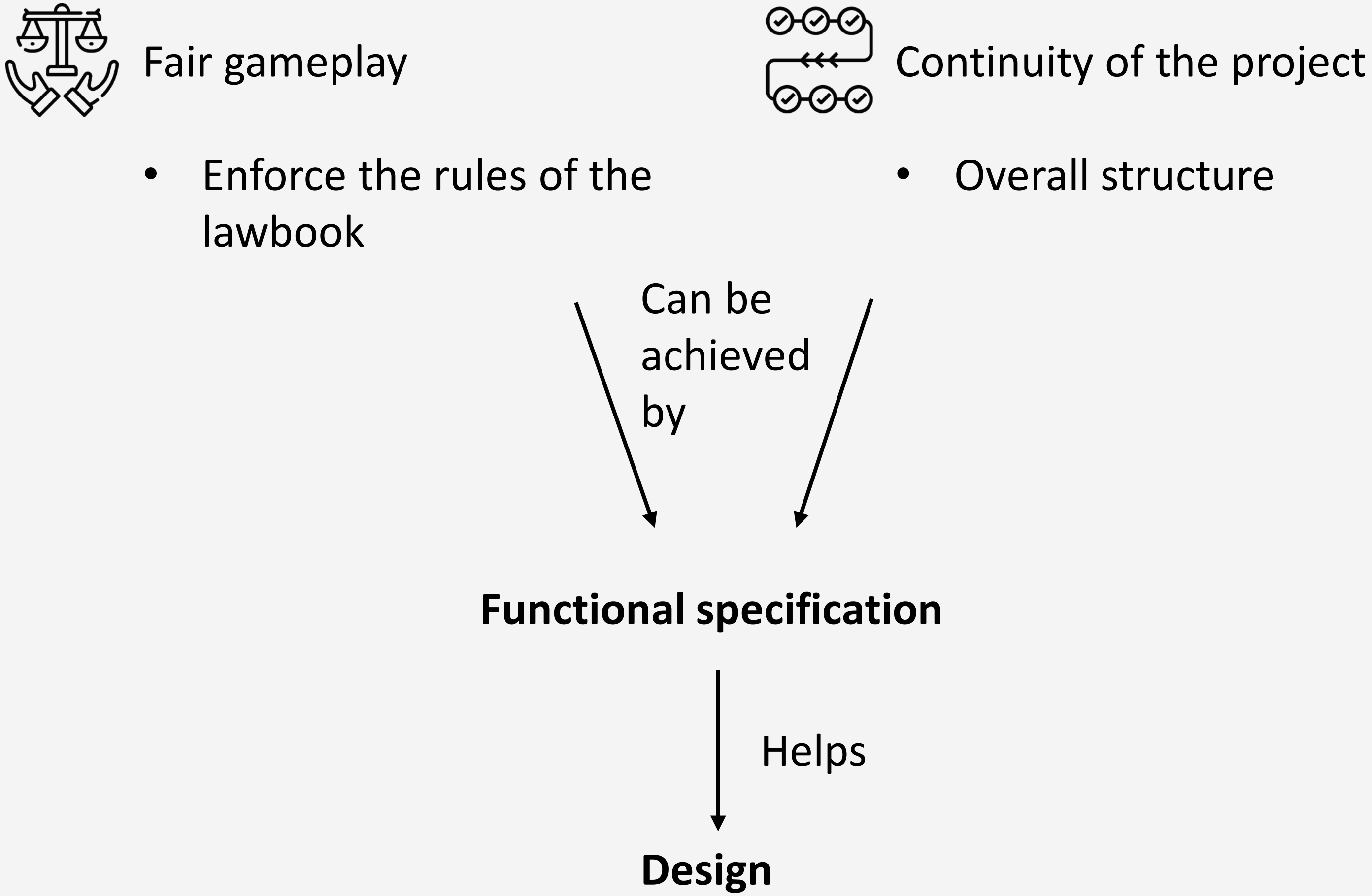
Fair gameplay

Continuity of the project



-University -RoboCup MSL	-TechUnited -Erjen -Rene
-previous MSD -Spectators	-Next MSD -Soccer teams -Referee, linesmen

2 Approach



Fair gameplay

- Enforce the rules of the lawbook

Continuity of the project

- Overall structure

Can be achieved by

Functional specification

Helps

Design

2 Approach

Project scope:

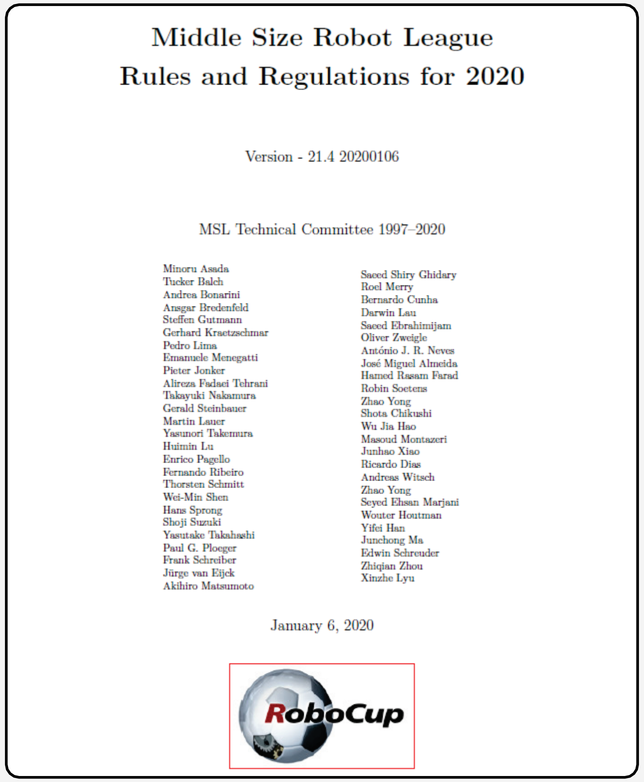
Functional specification development

Distance violation check task design

Past works archive

3 Functional Specification

Enforce the rules of the lawbook



Middle Size Robot League
Rules and Regulations for 2020

Version - 21.4 20200106

MSL Technical Committee 1997–2020

Minoru Asada
Tucker Balch
Andrea Bonarini
Ansgar Bredenfeld
Steffen Gutmann
Gerhard Kraetzschmar
Pedro Lima
Emanuele Menegatti
Pieter Jonker
Alireza Fadaei Tehrani
Takayuki Nakamura
Gerald Steinbauer
Martin Lauer
Yasunori Takemura
Huimin Lu
Enrico Pagello
Fernando Ribeiro
Thorsten Schmitt
Wei-Min Shen
Hans Sprong
Shoji Suzuki
Yasutake Takahashi
Paul G. Ploeger
Frank Schreiber
Jürge van Eijck
Akihiro Matsumoto

Saeed Shiry Ghidary
Roel Merry
Bernardo Cunha
Darwin Lau
Saeed Ebrahimijam
Oliver Zweigle
António J. R. Neves
José Miguel Almeida
Hamed Rasam Farad
Robin Soetens
Zhao Yong
Shota Chikushi
Wu Jia Hao
Masoud Montazeri
Junhao Xiao
Ricardo Dias
Andreas Witsch
Zhao Yong
Seyed Ehsan Marjani
Wouter Houtman
Yifei Han
Junchong Ma
Edwin Schreuder
Zhiqian Zhou
Xinzhe Lyu

January 6, 2020

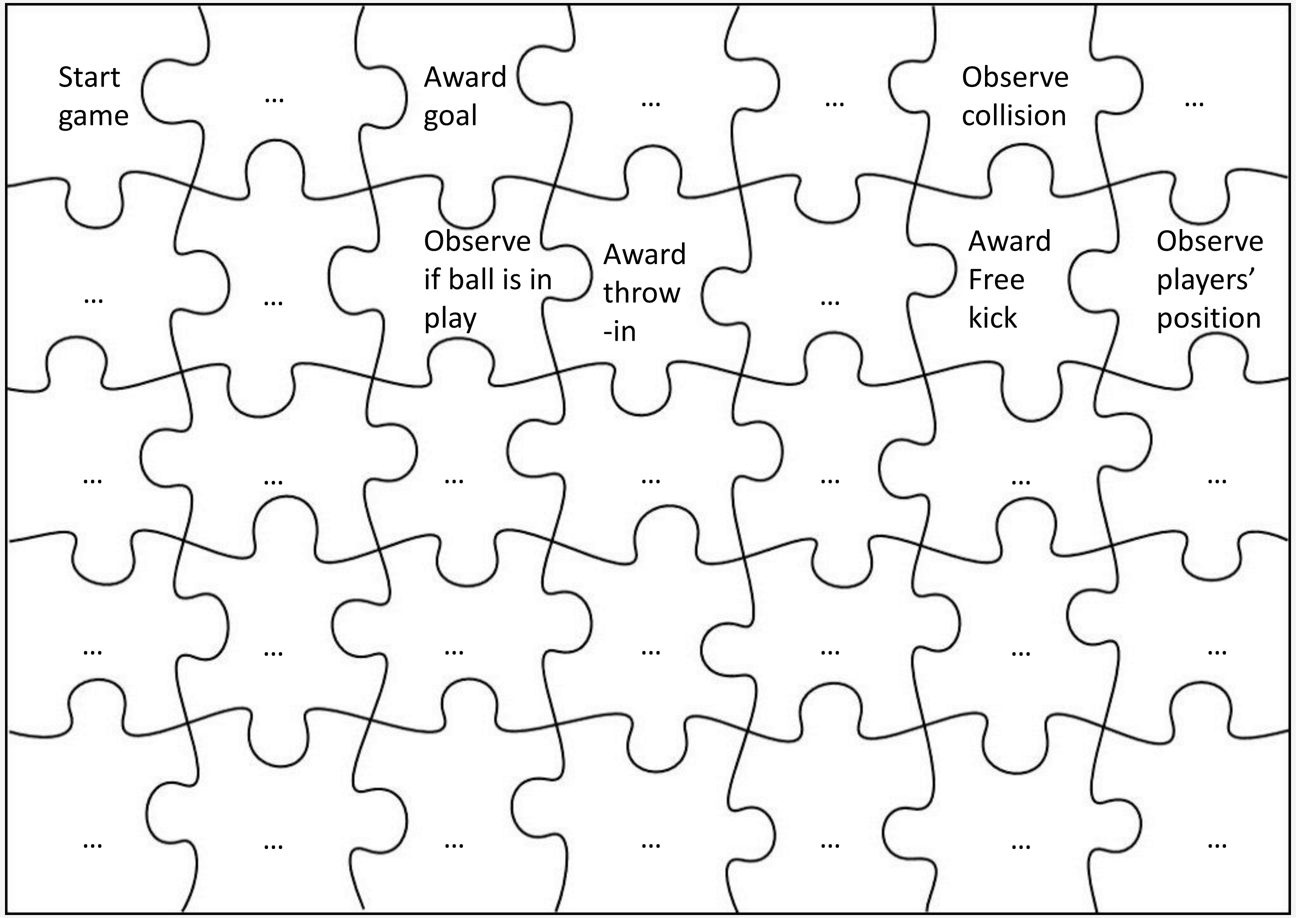
RoboCup

What does the referee do?

Functional specification

3 – Functional Specification

- **What is the functional specification?**



Start game	…	Award goal	…	…	Observe collision	…
…	…	Observe if ball is in play	Award throw -in	…	Award Free kick	Observe players’ position
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…

3 – Functional Specification

- **How are the functions specified?**

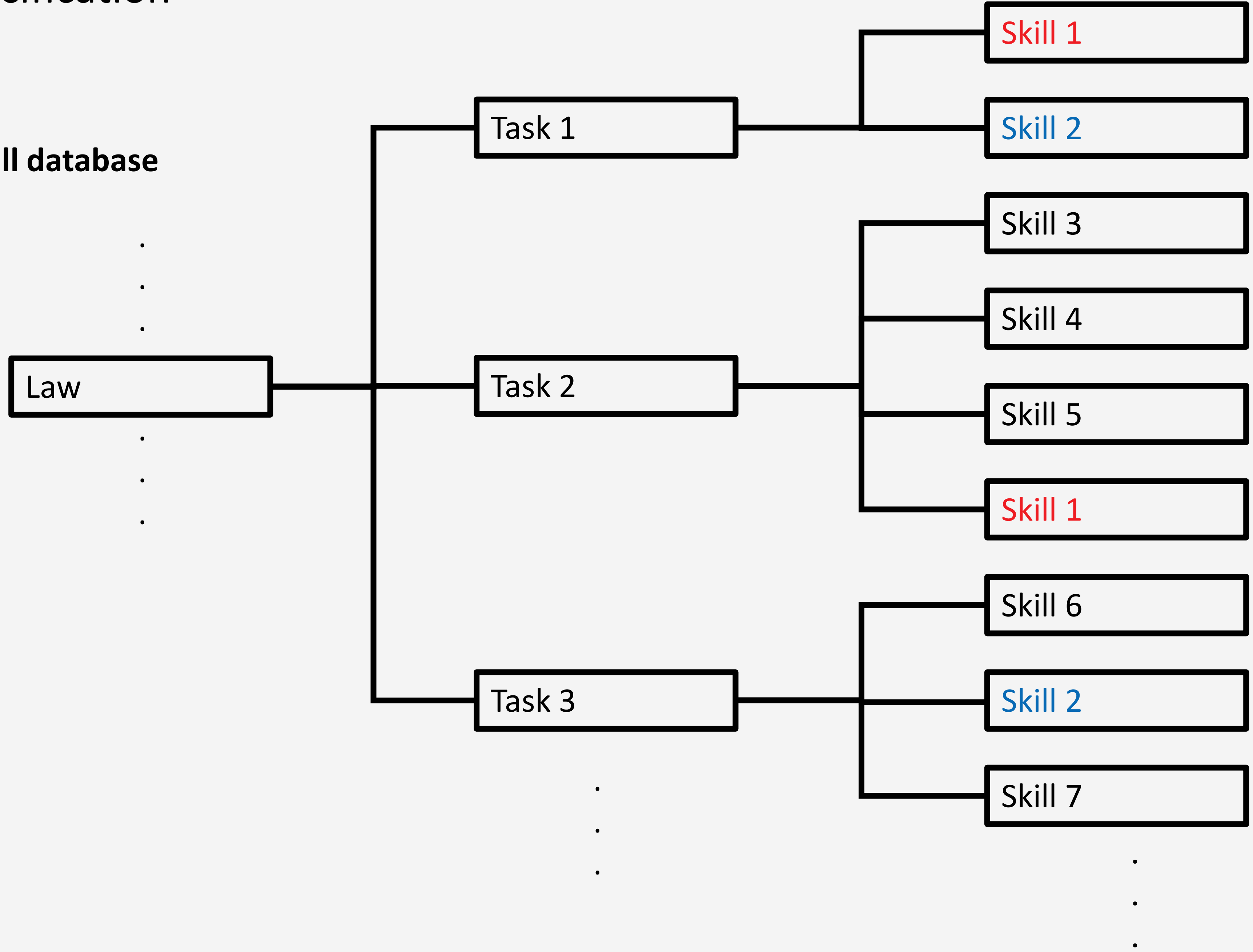
- Law → Task → Skill database

- Game-state flow visualization

3

Functional Specification

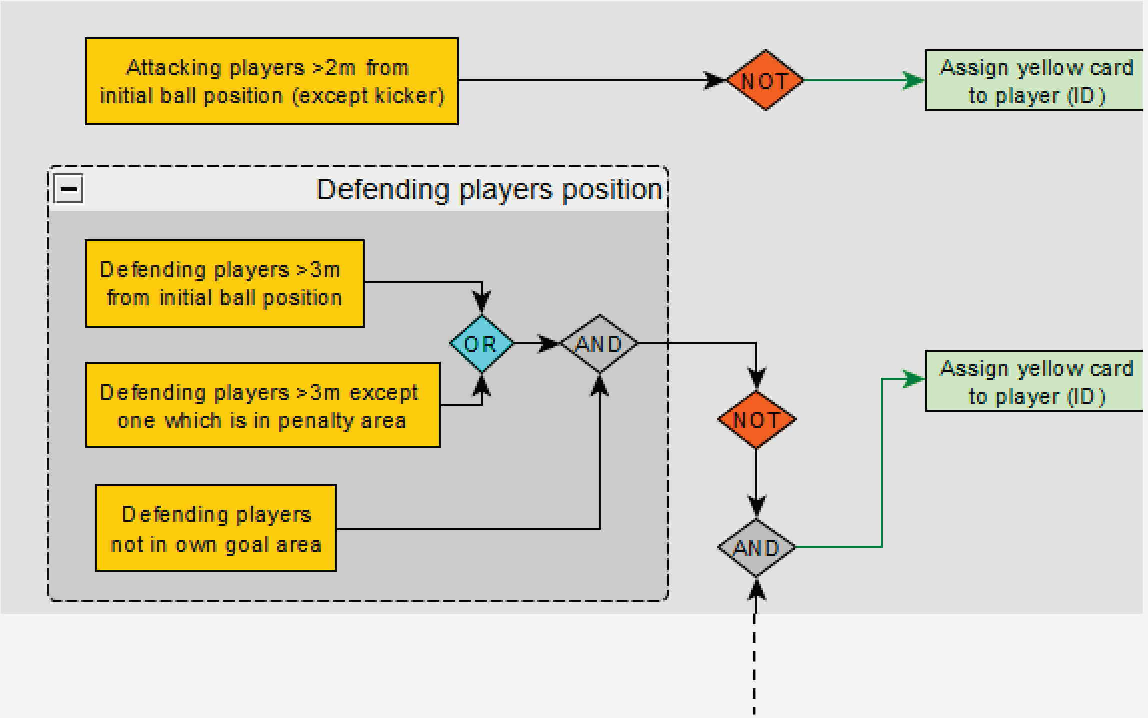
- **Law → Task → Skill database**



3 – Functional Specification

- **Game-state flow visualization**
  - Over arching game states
  - State specific referee tasks
  - Flow of the tasks

Example: Throw-in



Attacking players >2m from initial ball position (except kicker)
NOT
Assign yellow card to player (ID)
Defending players position
Defending players >3m from initial ball position
OR
AND
Defending players >3m except one which is in penalty area
Defending players not in own goal area
NOT
AND
Assign yellow card to player (ID)

Objective

Detect ball-player distance violation during the following game states:

1. Free kick
2. Kick-off
3. Corner kick
4. Goal kick
5. Throw-in
6. Penalty kick

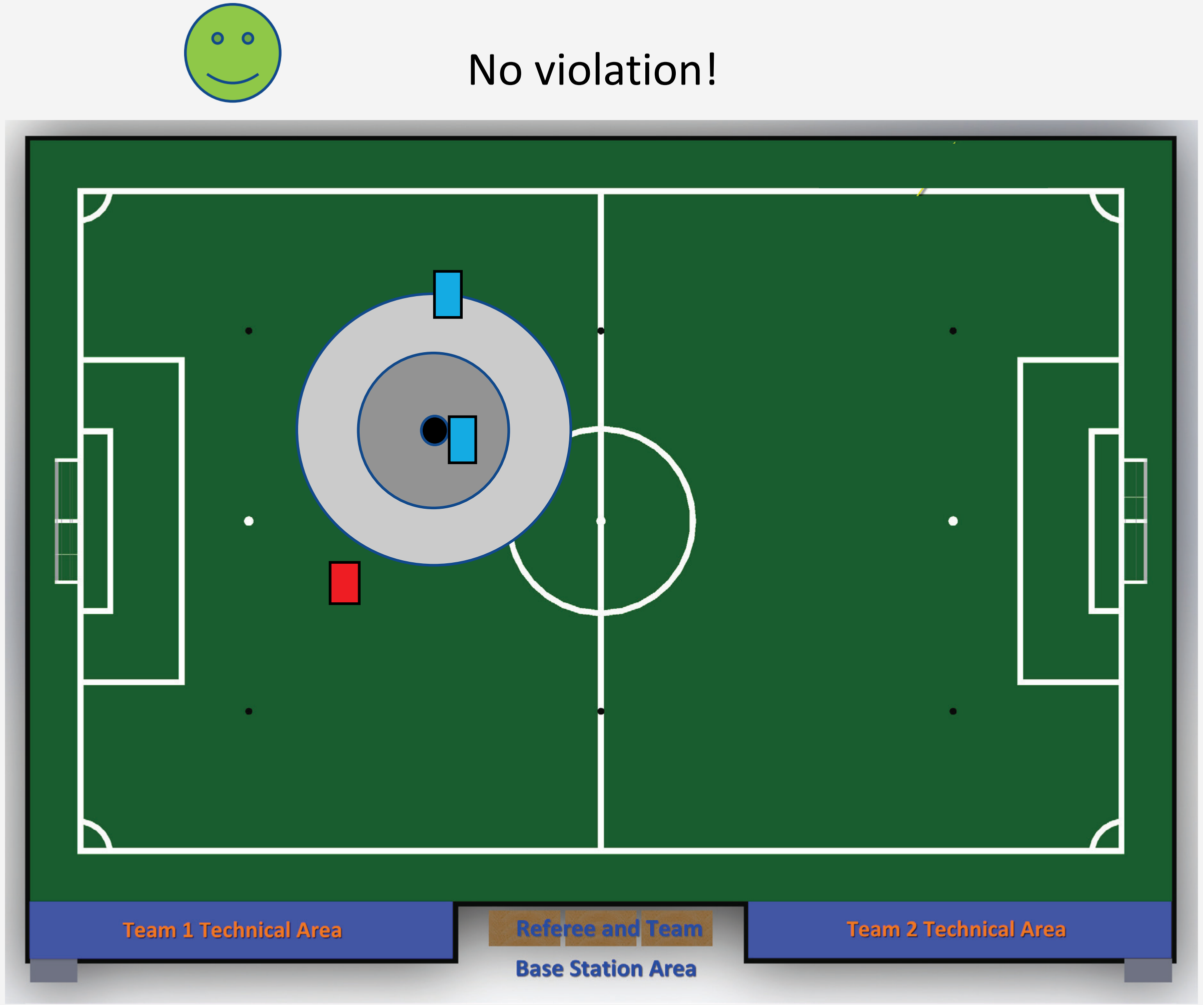
Objective

Detect ball-player distance violation during the following game states:

1. Free kick
2. Kick-off
3. Corner kick
4. Goal kick
5. Throw-in
6. Penalty kick

Free kick—Team Blue

No violation!



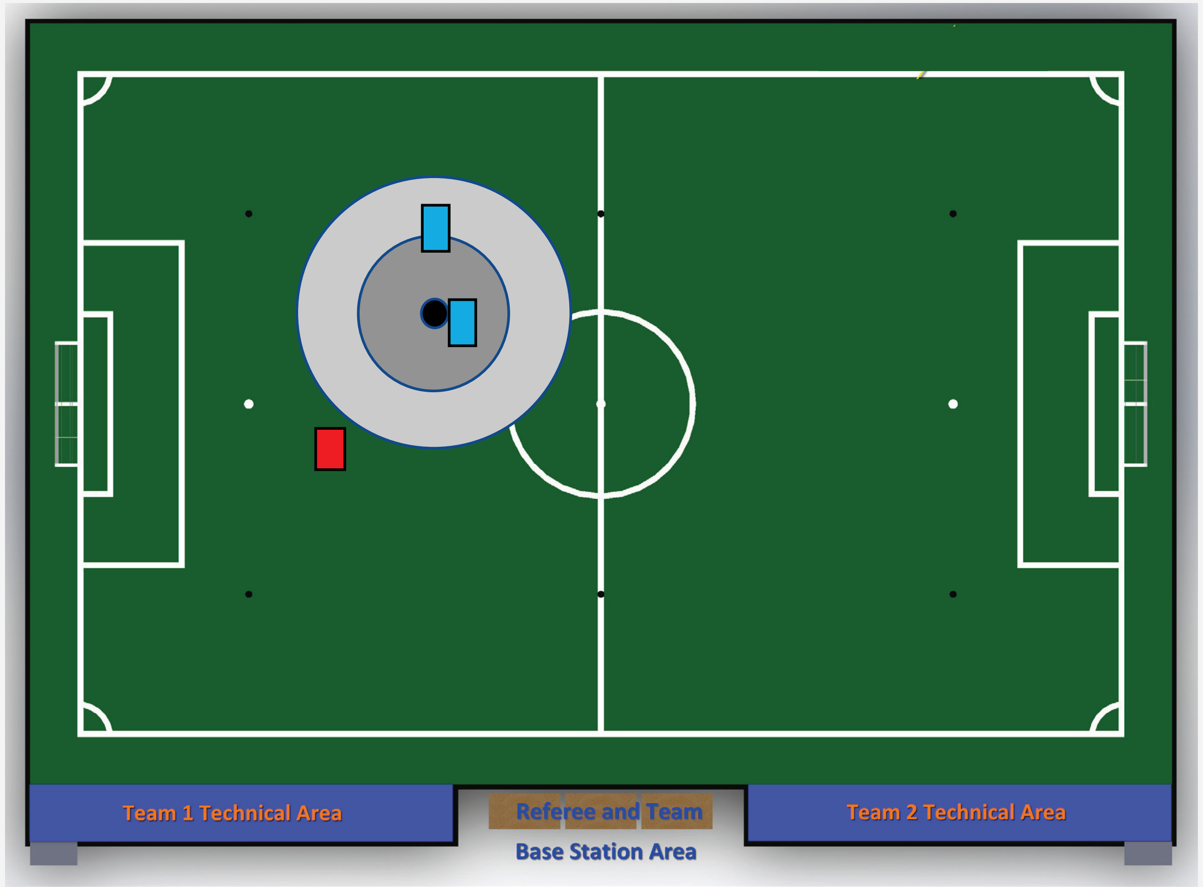
Objective

Detect ball-player distance violation during the following game states:

1. Free kick
2. Kick-off
3. Corner kick
4. Goal kick
5. Throw-in
6. Penalty kick

Free kick—Team Blue

🚫 Violation by team Blue



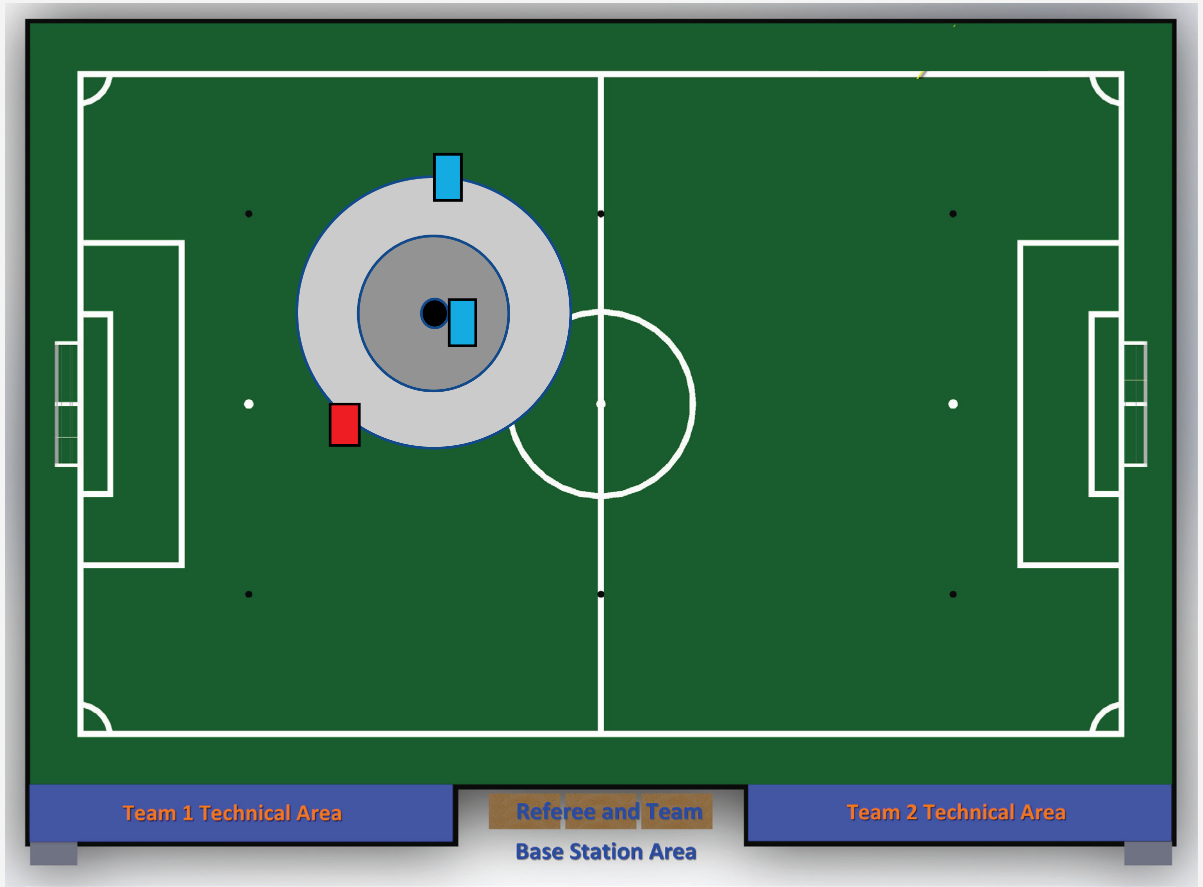
Objective

Detect ball-player distance violation during the following game states:

1. Free kick
2. Kick-off
3. Corner kick
4. Goal kick
5. Throw-in
6. Penalty kick

Free kick—Team Blue

🚫 Violation by team Red



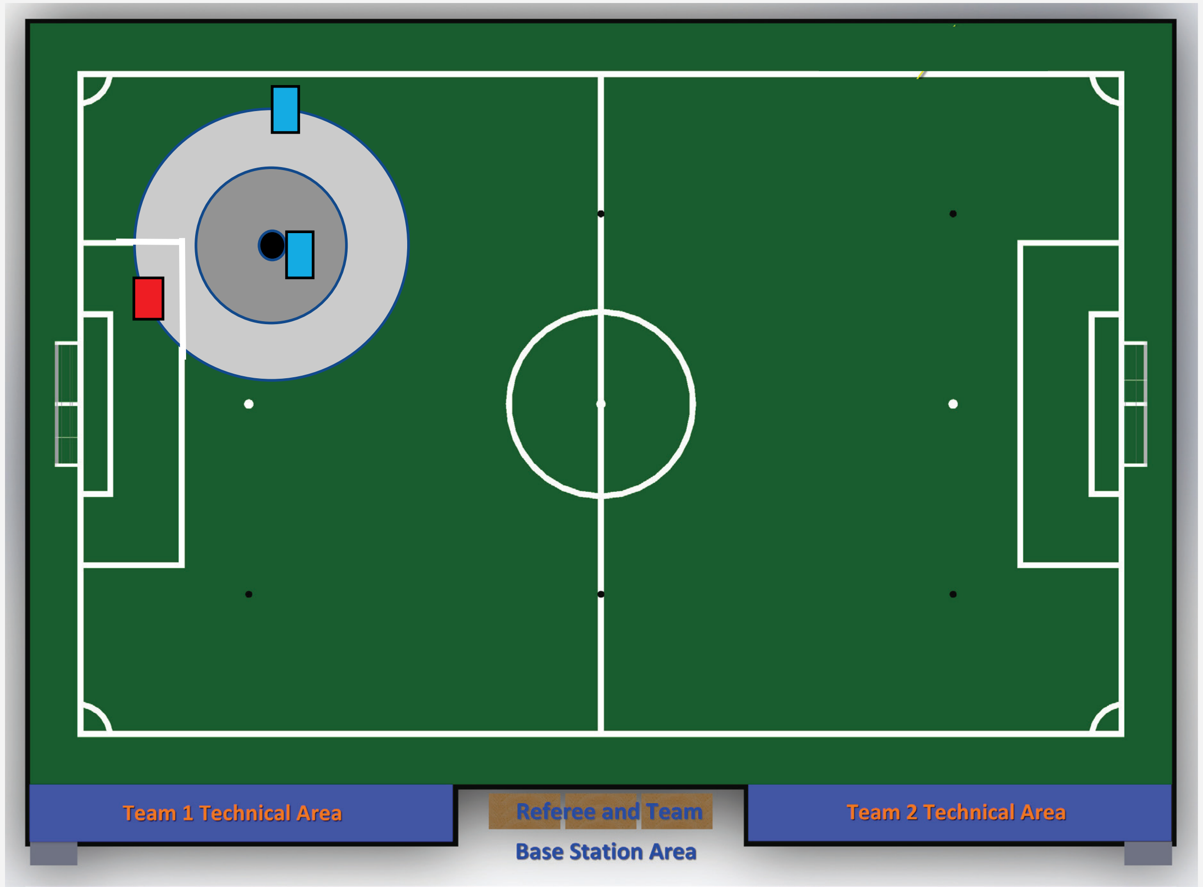
Objective

Detect ball-player distance violation during the following game states:

1. Free kick
2. Kick-off
3. Corner kick
4. Goal kick
5. Throw-in
6. Penalty kick

Free kick—Team Blue

No violation by team Red!
Penalty box exemption!



Motivation

New functionality for the system

Hard to realize for human-being

Proof of concept for functional specification

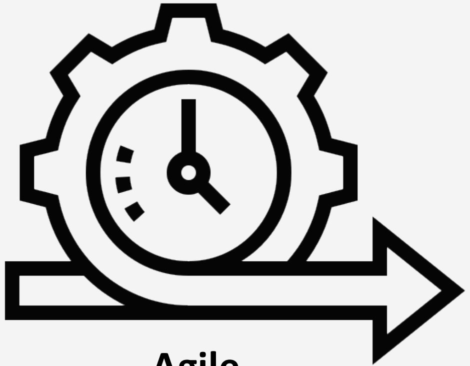
Corresponding to the team learning goals

Scope of design work

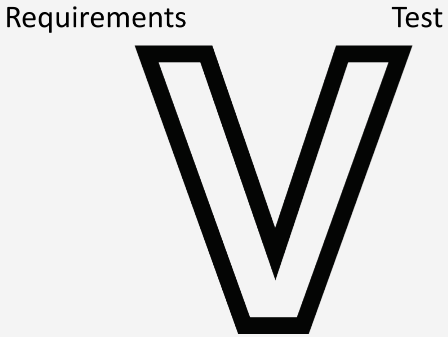
- Requirements formulation
- Algorithm architectural decomposition development
- Individual software blocks development
- Individual software blocks integration
- Algorithm testing on images and videos

4 Design

Approach



Agile



V–model

Major Design Choices

Programming language:

MATLAB®

Simulation environment:

SIMULINK®

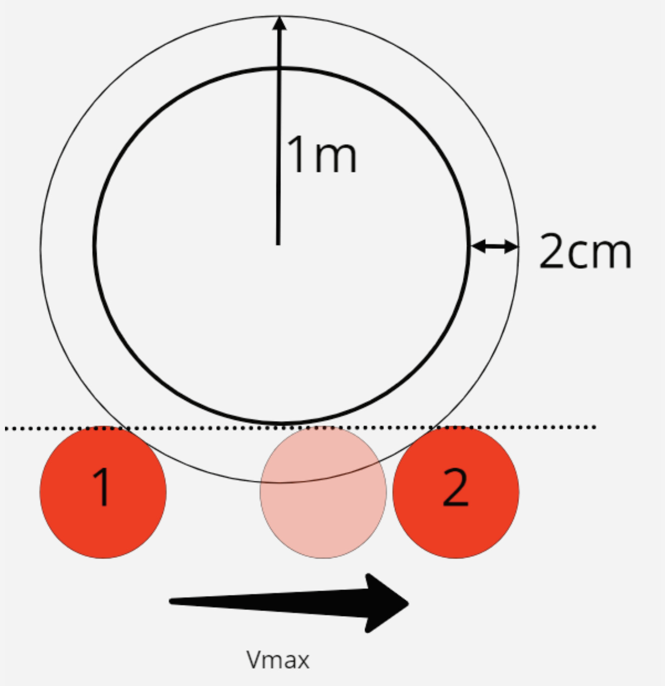
Vision System Parameters:

- Height – 12 m
- Frame of view (FOV) – 1.2 radians
- Resolution – 1920x1080

Requirements. Functional ones

- The system must **detect** the **players** and the **ball** inside the **soccer field** boundaries and identify the players’ team.
- The system must **detect** the different **zones** inside the soccer field (corner area, penalty area, etc.)
- The system must **check** if the **distance** between the center of the **ball** and any part of the **attacker team** members (except for the kicker) before free kick, corner kick, kick-off, goal kick, and throw-in is not less than **2m**. (with acceptable 5cm inaccuracy). **One** of the robots may stay anywhere inside the **penalty area** (except for the goal area) of its own team, even if the distance to the ball is shorter than 2m.
- The system must **check** if the **distance** between the center of the **ball** and any part of the **defender team** members before free kick, corner kick, kick-off, goal kick, and throw-in is not less than **3m**. (with acceptable 5cm inaccuracy). One of the defender robots may stay anywhere inside the **penalty area** (except for the goal area) of its own team, even if the distance to the ball is shorter than 3m.
- The system must **check** if the **distance** between the center of the **ball** and any part of all the **players** before the dropped-ball is **1m**. (with acceptable 5cm inaccuracy in this distance.). One of the robots may stay anywhere inside the penalty area (except for the goal area) of its own team, even if the distance to the ball is shorter than 1m.

Requirements. Performance. Frequency



The system must be able to **realize** the functional requirements (based on the system accuracy) at least every **89 ms** in order to avoid false-negative detections, which means it should have a **detection frequency** of **11.2Hz**.

Requirements. Color detection

The system must detect and distinguish objects mentioned in the Table by means of their RGB values (with ~8% margin for each channel value)

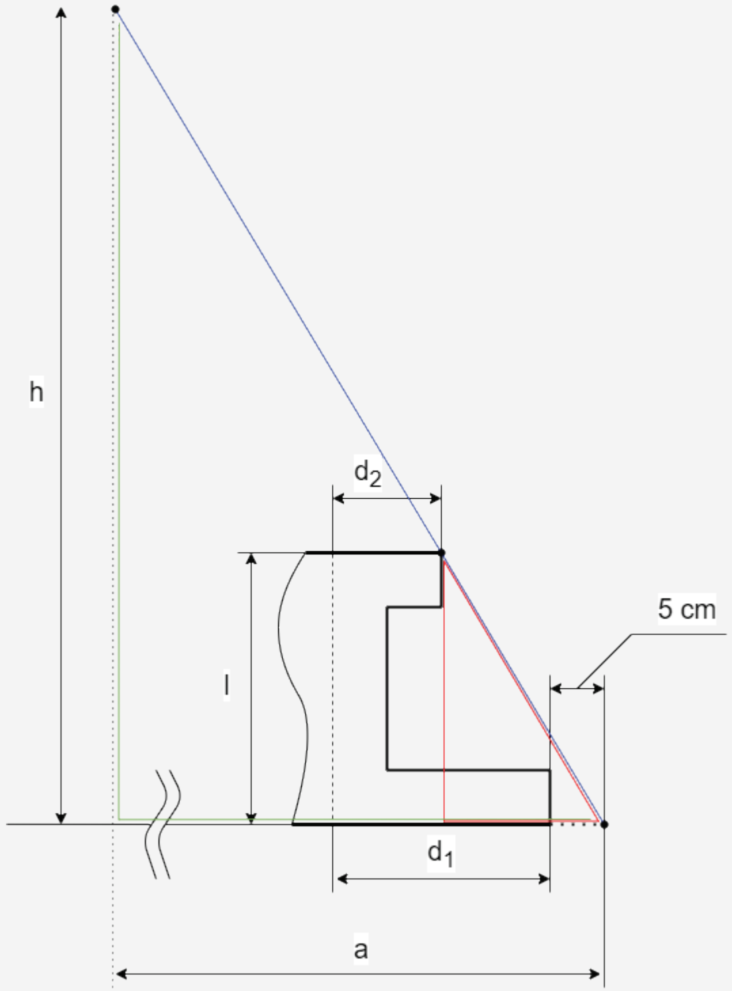
Object	RGB value
Ball	[255, 175, 10]
Team A player	[240, 10, 10]
Team B player	[250, 250, 10]

Requirements. Minimal distortion

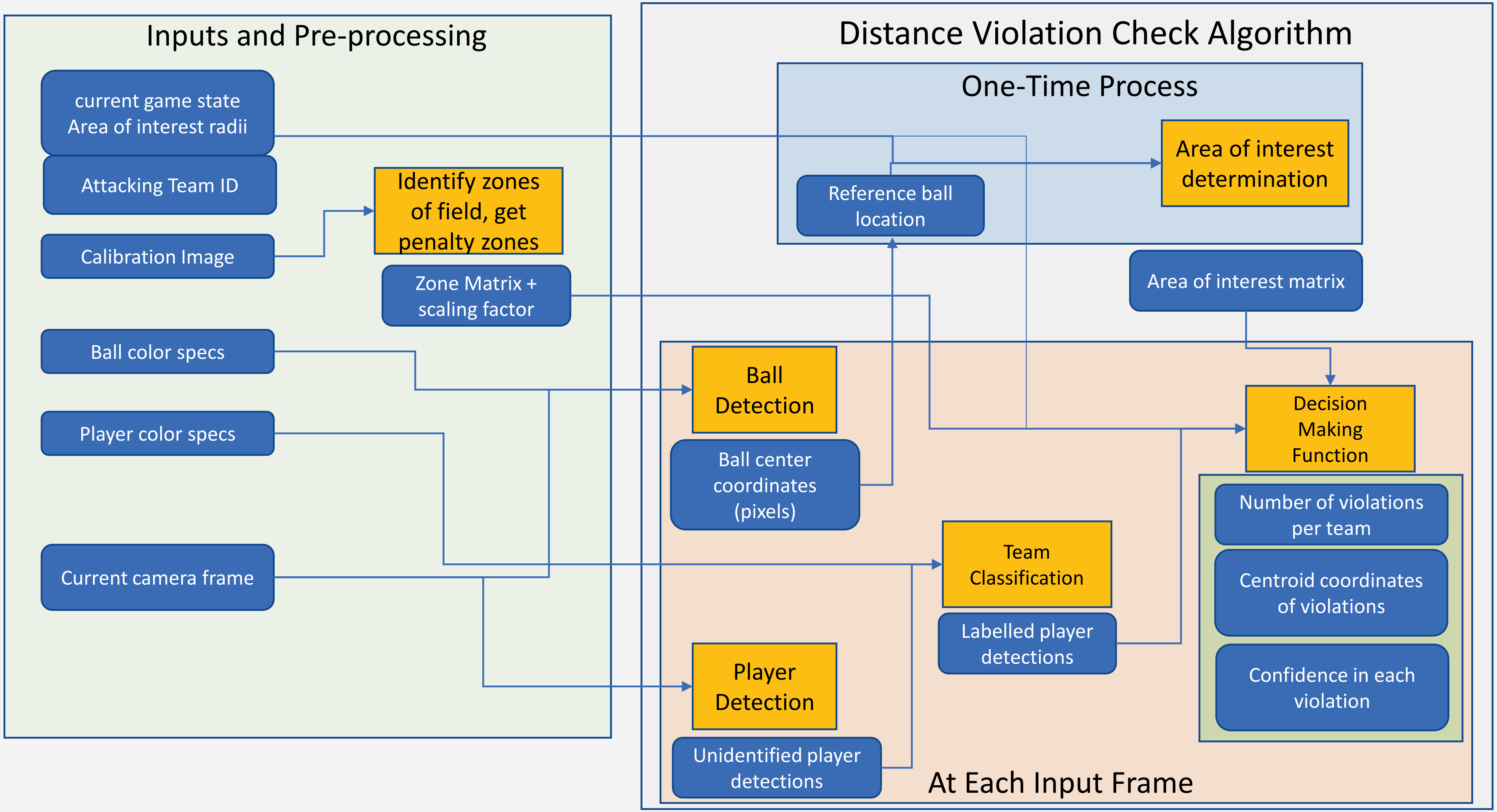
The system should capture an image of the furthest player in such a way that it’s top outermost point should not overcover the region of radius of it’s bottom outermost point with addition of 5 cm.

Otherwise it will lead to false positive

❗ The camera should be located at 150 meters height which is not feasible in real life



Software Decomposition



Inputs and Pre-processing

current game state
Area of interest radii

Attacking Team ID

Calibration Image

Identify zones of field, get penalty zones

Zone Matrix + scaling factor

Ball color specs

Player color specs

Current camera frame

Distance Violation Check Algorithm

One-Time Process

Reference ball location

Area of interest determination

Area of interest matrix

Ball Detection

Ball center coordinates (pixels)

Team Classification

Labelled player detections

Player Detection

Unidentified player detections

Decision Making Function

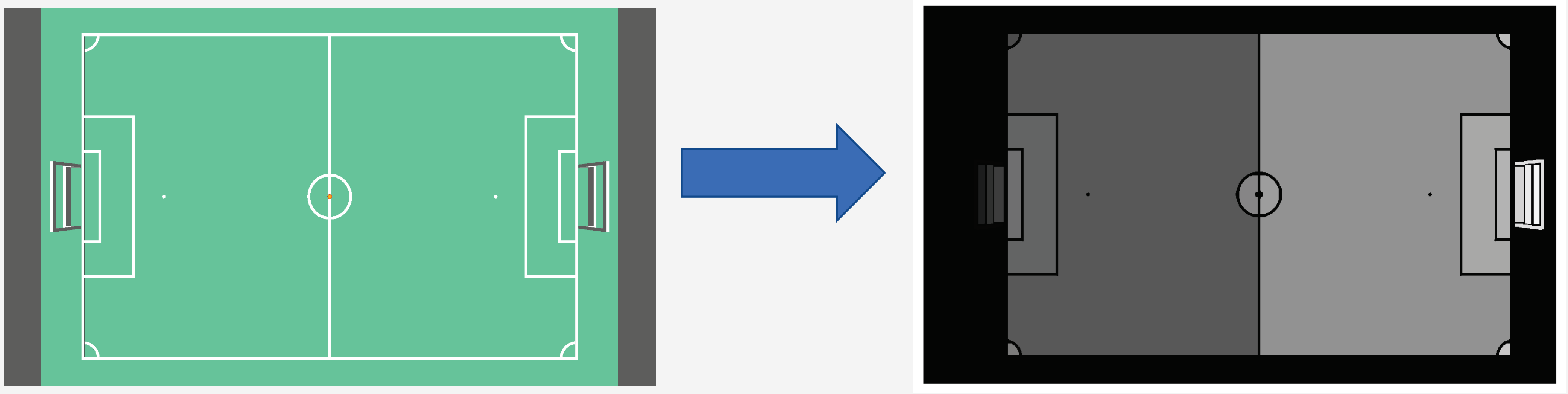
Number of violations per team

Centroid coordinates of violations

Confidence in each violation

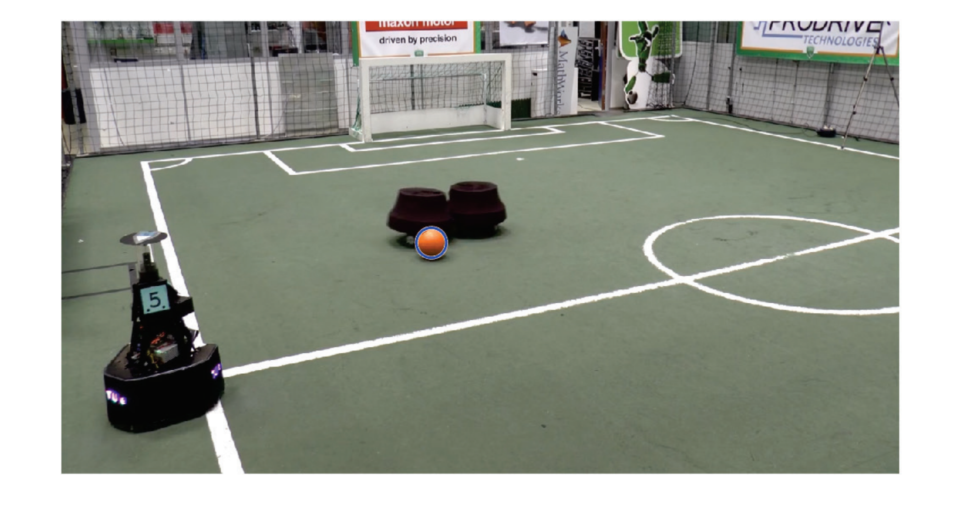
At Each Input Frame

Zone of Field Identification

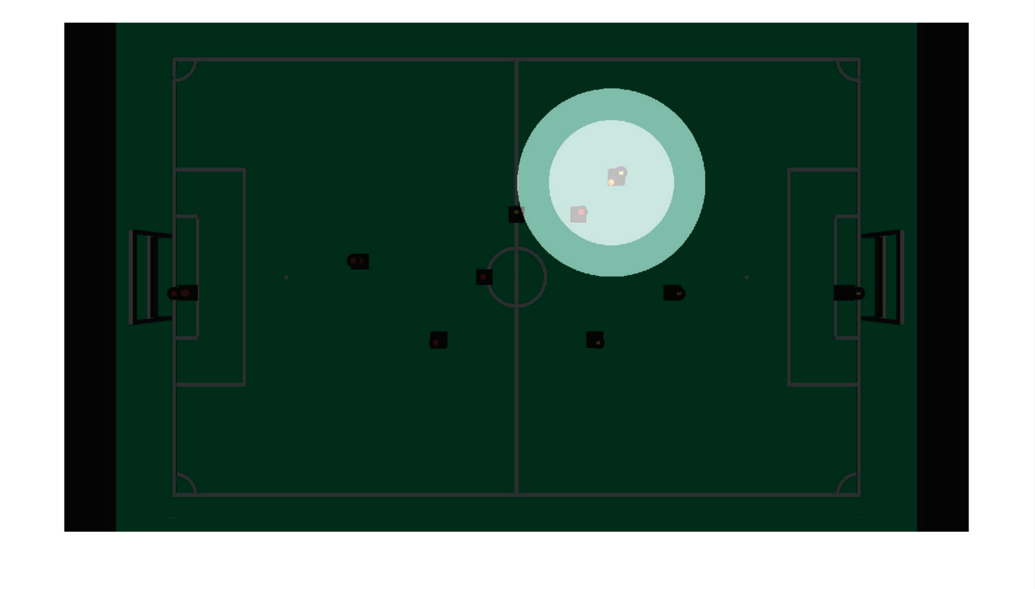


Ball detection and Area of interest identification

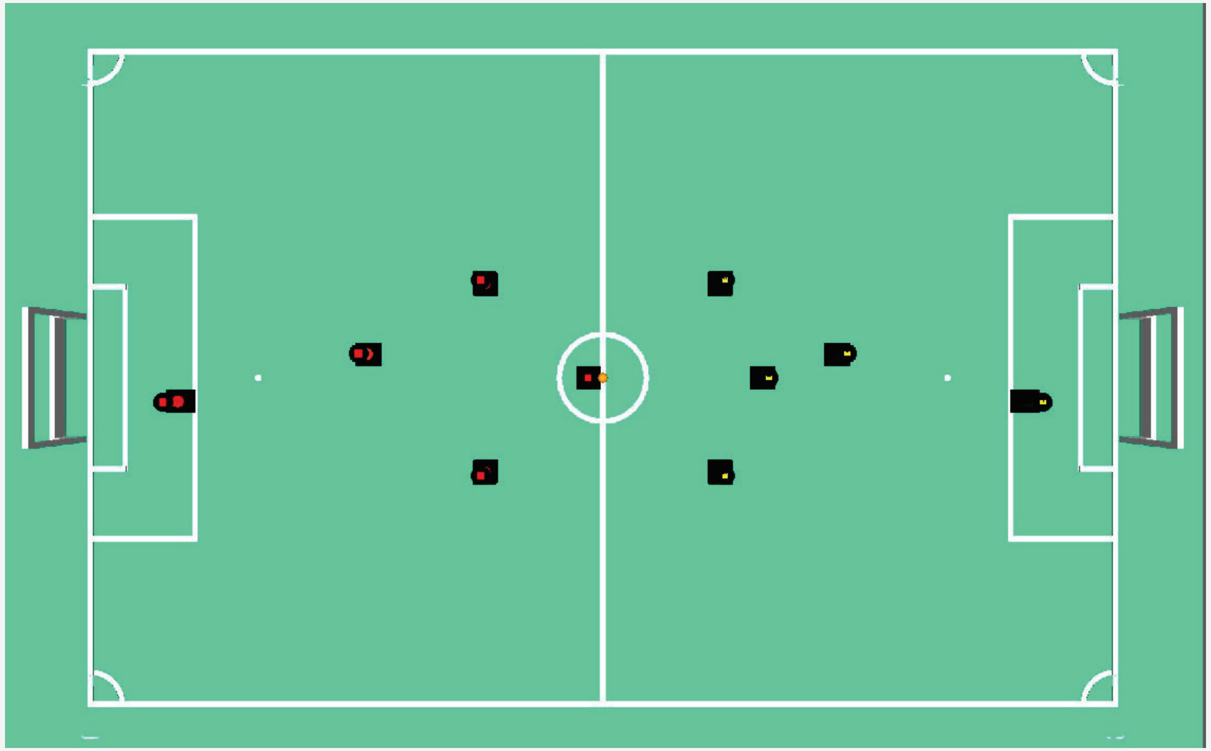
Ball Detection



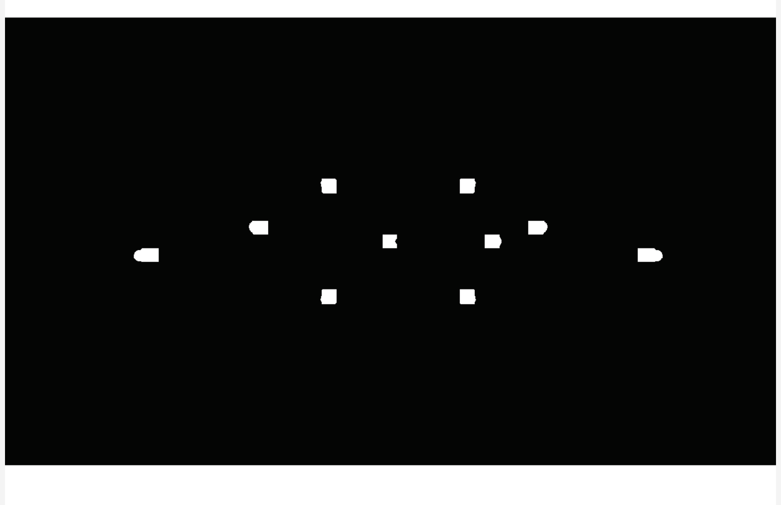
Area of interest identification

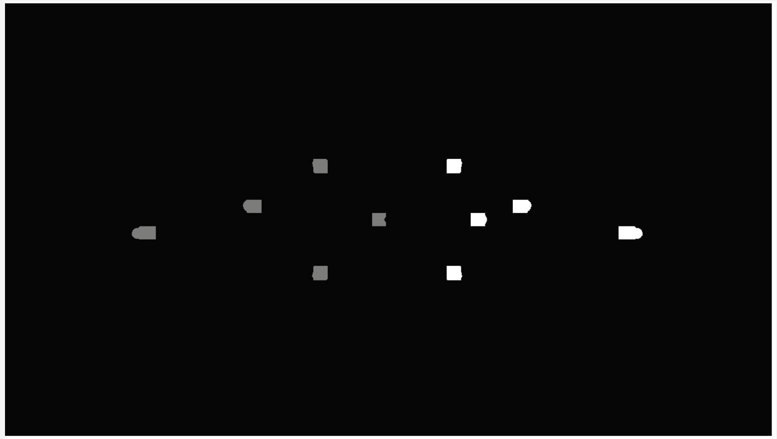


Player Detection and Classification



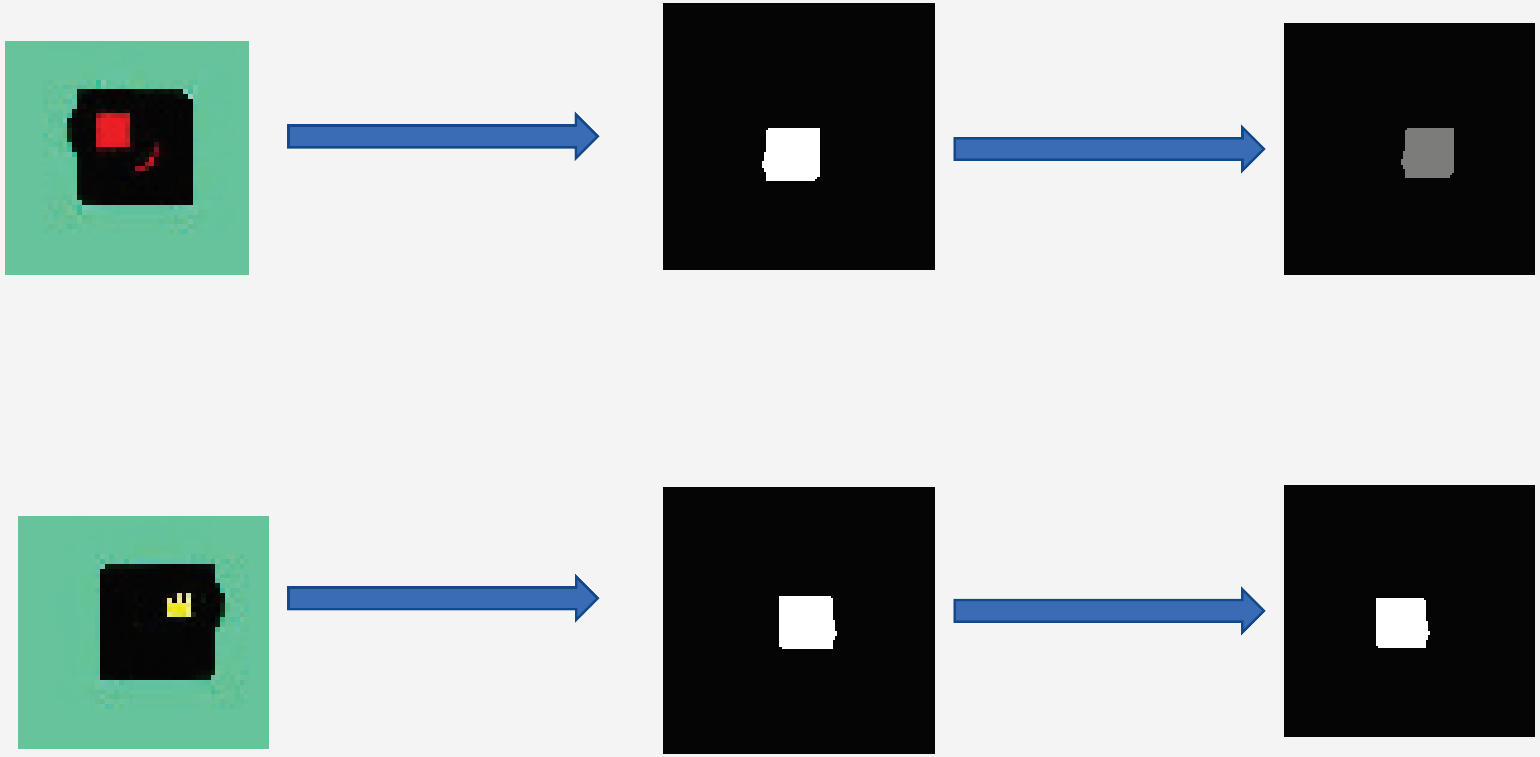
Unlabeled Player detections





Labelled Player detections

Player Detection and Classification



Decision Making Function

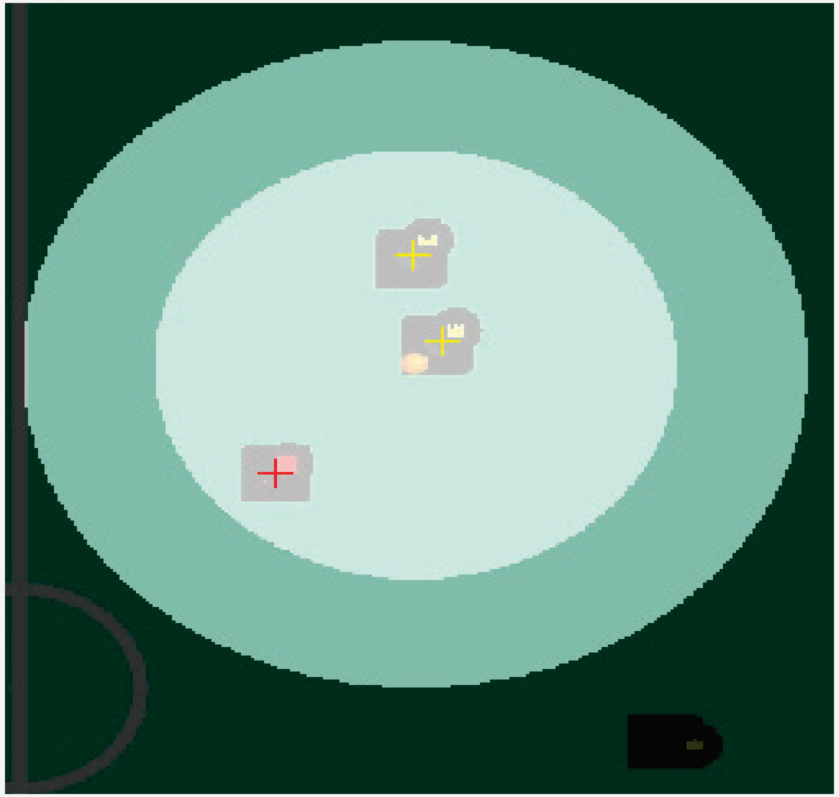
Inputs:

1. Kicker team and game state
2. Areas of interest
3. Player detection matrices
4. Labelled zone matrix

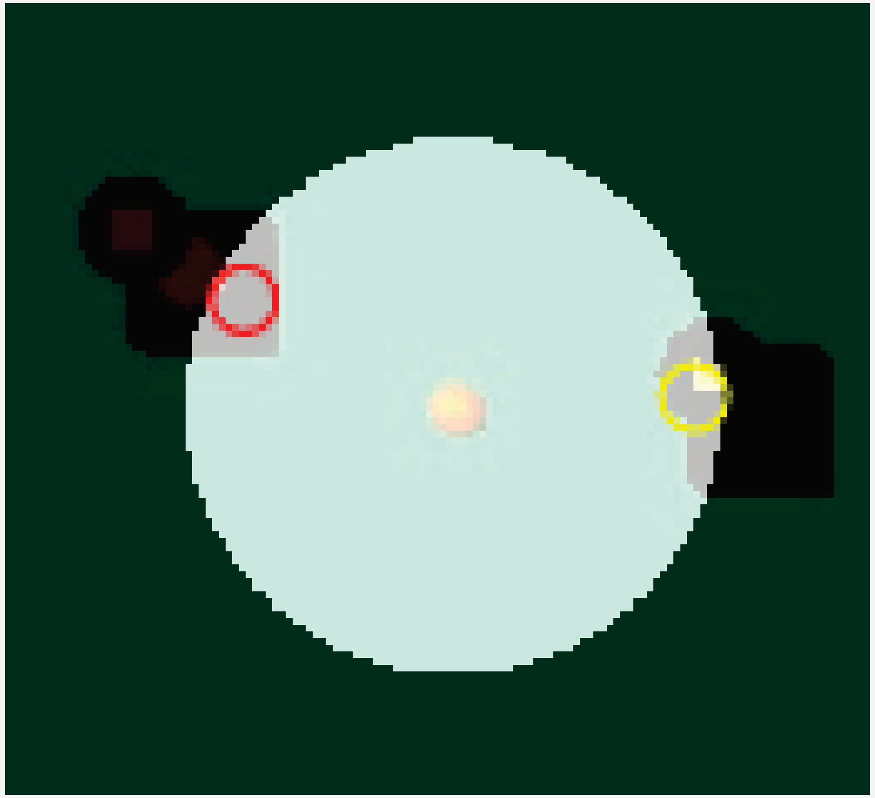
Outputs

1. Number of violations per team
2. Confidence in each violation

High Confidence violation



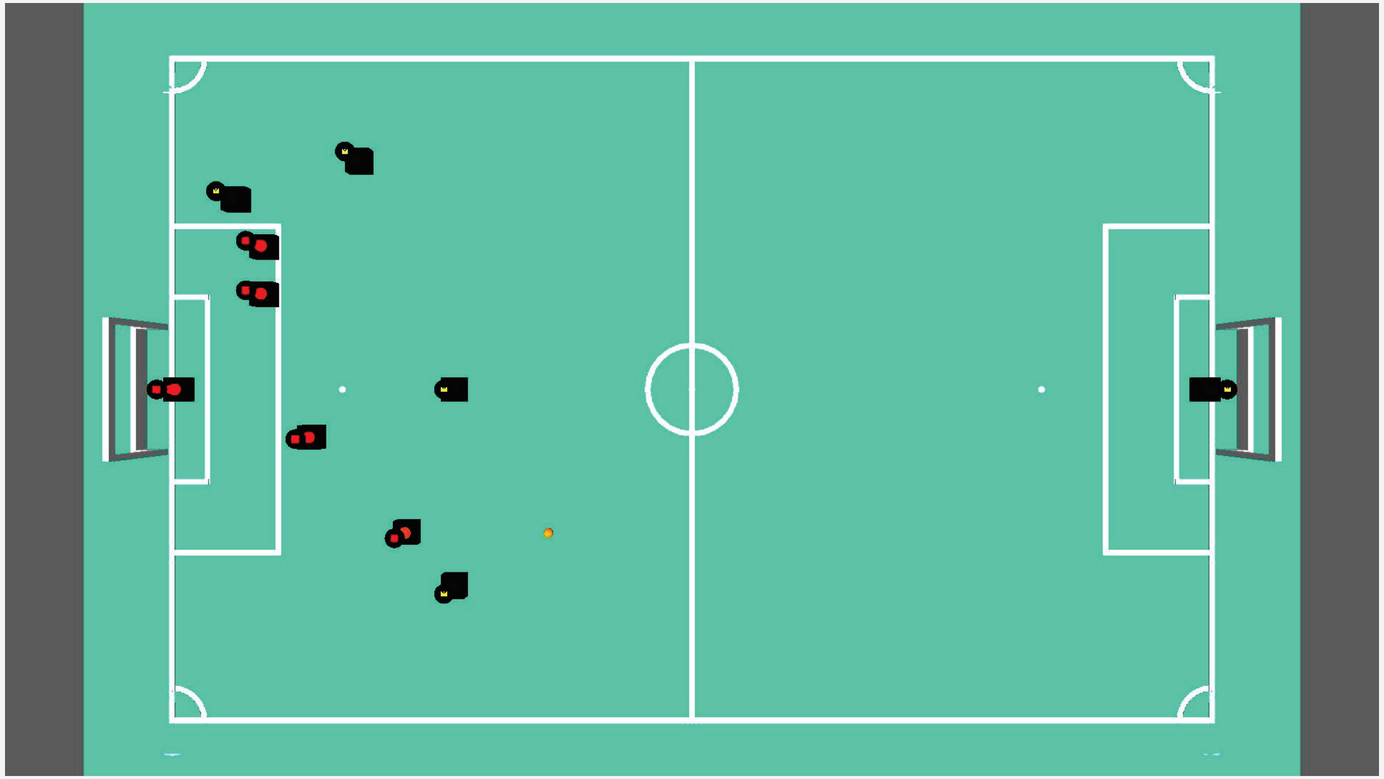
Low Confidence violation



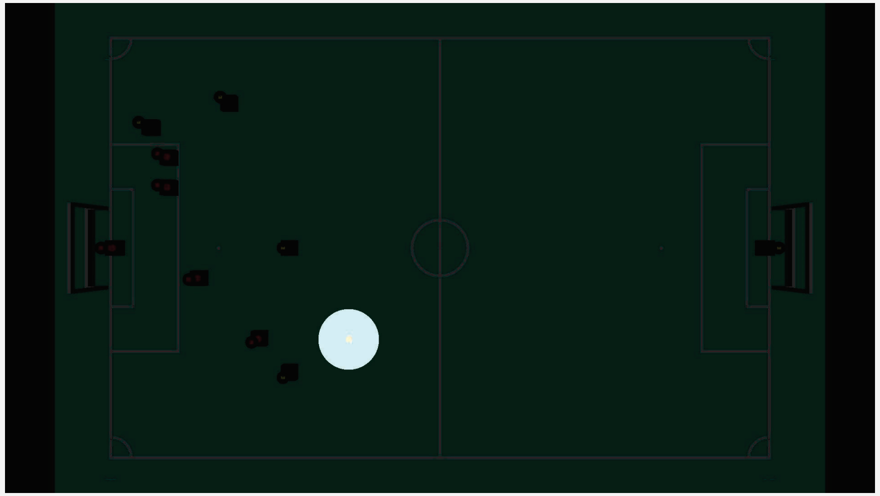
Video Tests

Extreme Case Scenario 2

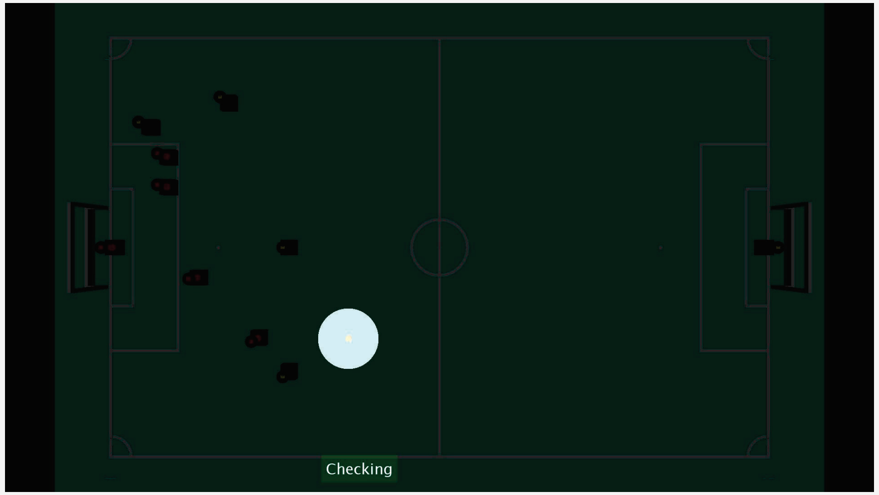
Scenario: Player tangentially passes through area of interest



Result: Independent of Algorithm speed



Result: Dependent on Algorithm speed



4 –Design

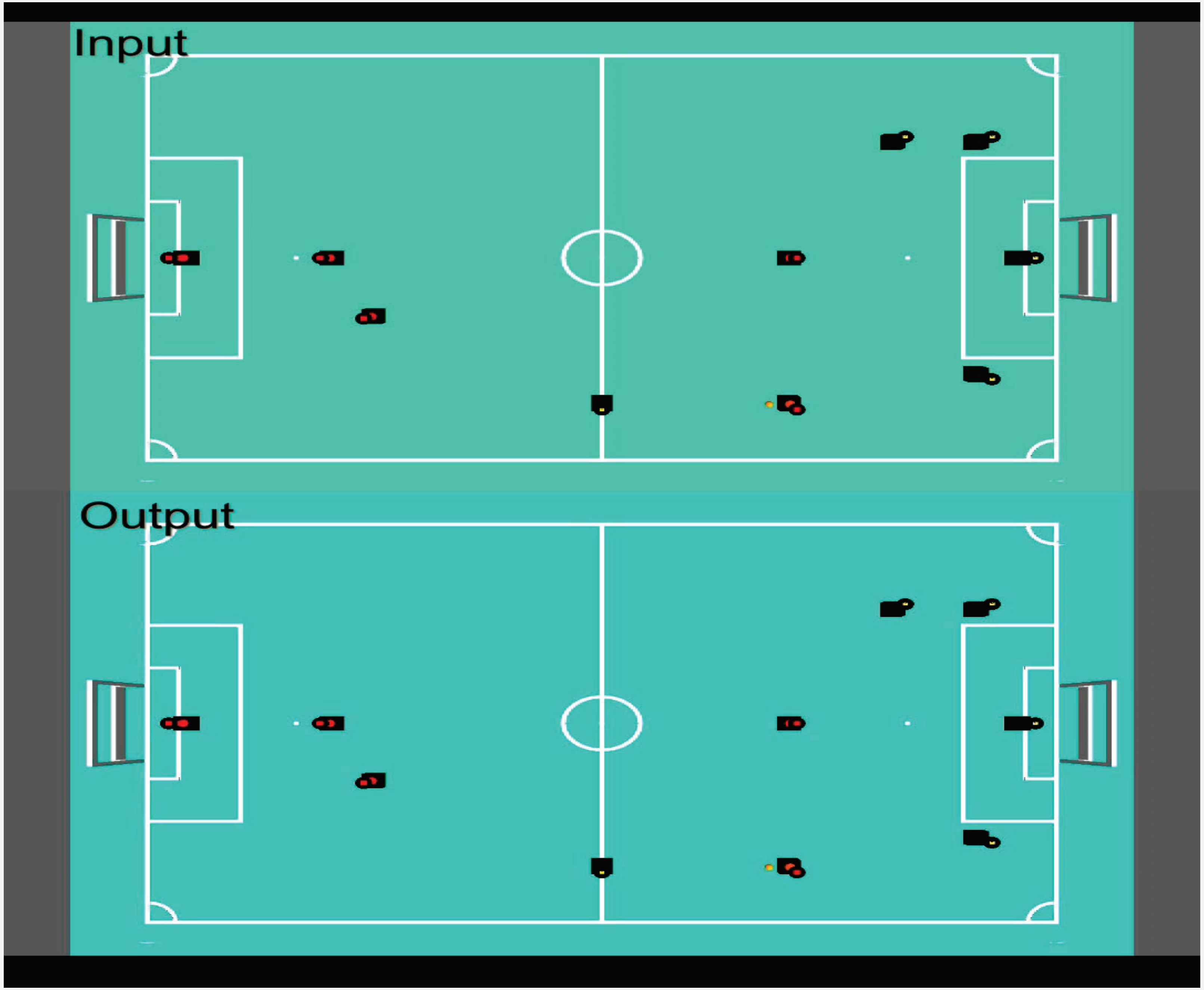
Main Highlights

- Straightforward implementation when using top camera system
- Use of confidence estimate on violations could give flexibility in enforcing decisions

Main Issues

- High algorithm execution time—current execution time is approximately 0.4 -0.5 seconds, (2 Hz), below the required specification of 12 Hz
- Dealing with occlusions– if the ball is hidden from view, last known position is considered
- Separating ‘connected players’– additional camera viewpoints (from sides) would be useful to separate players

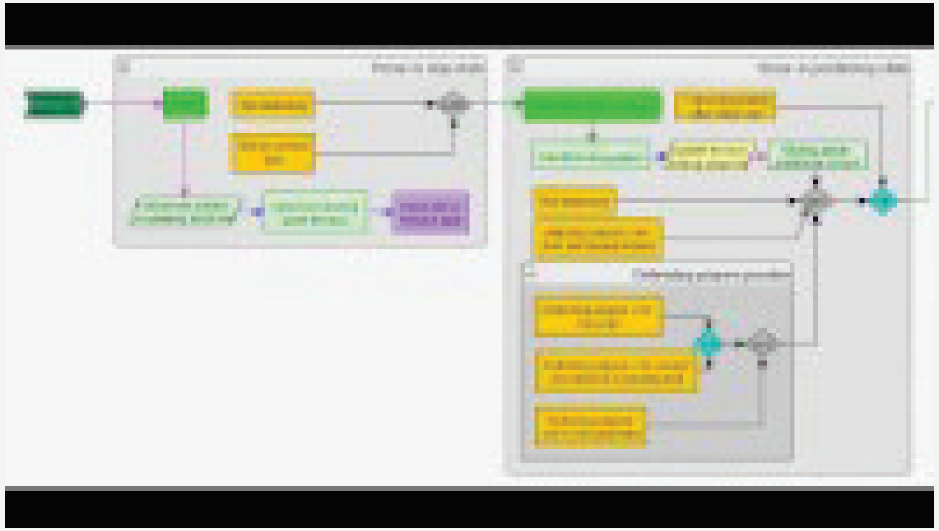
4 —Design



5 Conclusions

- **Functional specification**
  - Complete framework for functional specification
  - Part of the lawbook fully specified
  - Overall game-state framework

- **Distance violation design**
  - Top view static camera concept explored– good alternative to drone
  - Successful implementation in simulation environment
  - Key improvement points identified



- **Explanation video for future generations**

6

Future work

- **Functional specification**
  - For future system architecture
    - Fully specify the database and visualization
    - Synchronize and unify the database and visualization
  - For future implementations
    - Use the functional specification for next implementations

- **Distance violation design**
  - Integrate additional viewpoints to the top-camera system concept
  - Use C++ code during hardware implementation

- **Keep the archive up to date**

TU/e

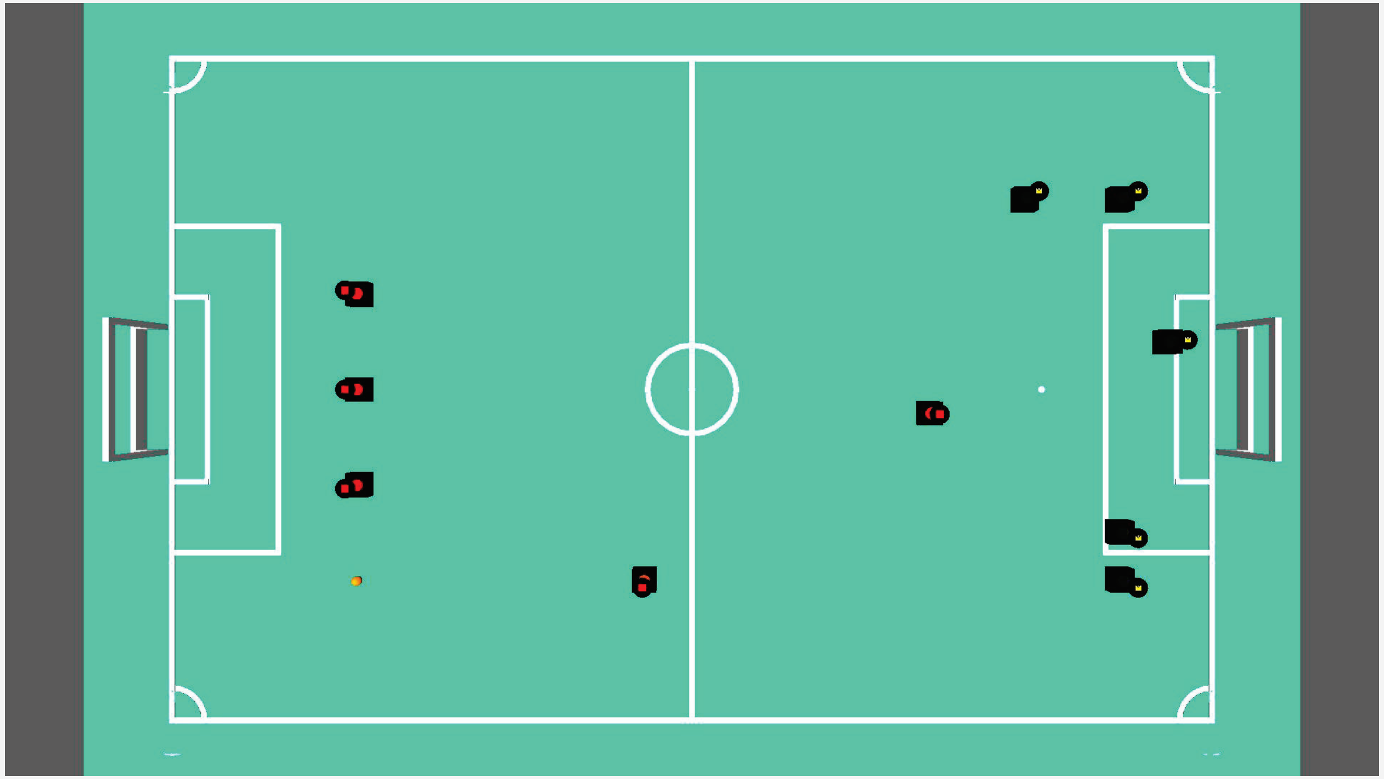
EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Thank you

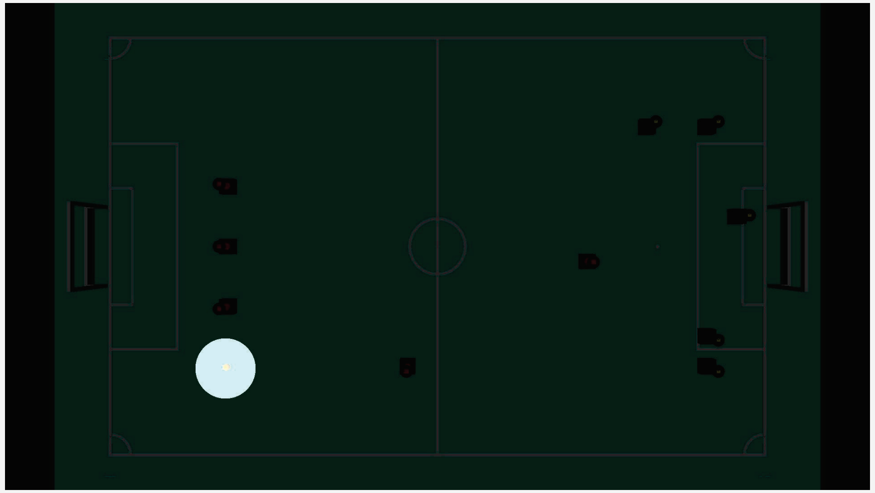
Video Tests

Extreme Case Scenario 1

Scenario: Player enters area of interest and reverses



Result: Independent of Algorithm speed



Result: Dependent on Algorithm speed