David António Freire Moura

Bachelor in Computer Science

Exploratory Analysis of Individual Metrics in Team Sports Using *IoT* Sensors

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Informatics Engineering

Adviser: Carmen Pires Morgado, Assistant Professor, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Co-adviser: Ruben Duarte Dias da Costa, Director, Knowledge Biz Consulting

Examination Committee

DRAFT: 2020-01-29

Exploratory Analysis of Individual Metrics in Team Sports Using Inertial Sensors

This document was created using the (pdf)LaTeX processor, based on the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Lorem ipsum.

Acknowledgements

The acknowledgements. You are free to write this section at your own will. However, usually it starts with the institutional acknowledgements (adviser, institution, grants, workmates, ...) and then comes the personal acknowledgements (friends, family, ...).

Abstract

Team Sports like basketball, football and baseball rely on data insights to improve player and team performance, practice scheduling and injury recovery. Recent improvements in data mining and machine learning techniques have encouraged the research of better ways to collect player data.

There are different approaches currently available, using video tracking systems or wearable sensors. The latter systems can be very precise in tracking position outdoors using GPS or indoors using Ultra-Wideband or RFID positioning, they fall short in the analysis of game related metrics, like accelerations and jumps or passes and shoots in the case of Basketball.

This work proposes a solution that can track both position and metrics of players and teams in real-time using inertial sensors, providing the users (e.g. players, coaches) an application with insightful information in order to improve the performance of the individual player and the whole team.

Resumo

Desportos colectivos tais como o basquetebol, futebol ou basebol usam dados para melhorar o desempenho em jogo, o planeamento de treinos e a recuperação de lesões de jogadores e da equipa. O aperfeiçoamento de técnicas como *data mining* e *machine learning* têm levado à pesquisa de formas de recolher mais e melhores dados dos jogadores.

Hoje em dia há várias abordagens diferentes: usando sistemas de captura de video ou utilizando sensores. Esta última abordagem pode ser bastante precisa no posicionamento em exteriores, utilizando sistemas como o GPS, ou em interiores, utilizando sistemas de posicionamento como Ultra-Wideband ou RFID. No entanto, estes sistemas não efetuam o reconhecimento de métricas de jogo como acelerações ou saltos, ou passes e lançamentos no basquetebol

Este trabalho propõe uma solução que visa medir em tempo real o posicionamento de jogadores e da equipa no campo, assim como recolher métricas de jogo, utilizando sensores de inércia. Deste modo, pretende-se fornecer aos utilizadores (e.g. jogadores, treinadores) uma aplicação com informações úteis e detalhadas sobre o desempenho dos jogadores individualmente, e da sua equipa como um todo.

Contents

List of Figures

List of Tables

Acronyms

Chapter 1

Introduction

1. Big picture dos desportos e tracking em desportos
2. Falar da monitorização de jogadores com câmaras e sensores
3. Problemas dessas abordagens.
4. Objetivos:
    - Maior precisão em métricas individuais usando wearables
    - Abordagem low-cost com IMUs
    - Solução wearable com processamento local e central, com vários sensores por jogador
    - Métricas a medir

Chapter 2

Related Work

Trabalho relacionado sobre:

1. Algoritmos de tracking de posição usando IMUs
    - Analysis of NBN23 system:
    - Madgwick
    - Carl Fischer
2. Algoritmos de métricas
    - 9 movement recognition with neural networks:
3. Outros trabalhos de métricas de performance (em equipa?)
4. ???Rede de Sensores???

Chapter 3

System Proposal

To achieve the goals set in Chapter **??**, a system that collects and processes the raw data from the IMU sensors and transforms it into meaningful metrics, storing it, and then presents it in a clear way to the user has to be built.

The proposed architecture is shown in Figure **??**, and it is comprised of three segments:

1. Edge - Data Collection and Processing
2. Server - Data Storing and Application Server
3. Client - Frontend Application

3.1 Edge

The Edge segment is composed of at least one IMU Sensor and one Raspberry Pi, connected via Bluetooth.

As the number of sensors can grow, and due to the limitation of the number of connections to a Bluetooth receiver, the number of Raspberry Pi's can also grow, in order to establish connection with all the IMU Sensors. This also helps to share the computation of the game metrics between the Raspberry Pi's.

3.1.1 IMU Sensors

The IMU sensors are attached to the players, and their only job is to send accelerometer and gyroscope raw data to the Raspberry Pi they are connected, over Bluetooth.

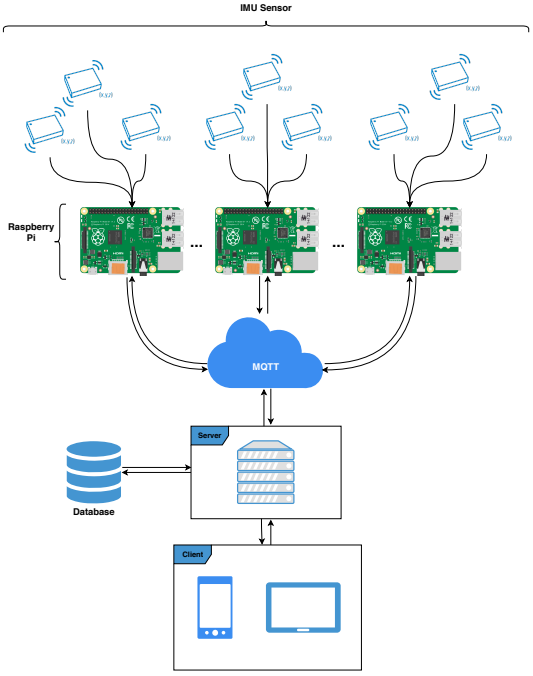

Figure 3.1: Proposed Architecture

3.1.2 Raspberry Pi

In the Raspberry Pi is where the "hard"work is done. Besides connecting and controlling the IMU Sensors, this unit has two main jobs: to collect the raw data sent by the IMU Sensors several times per second, and perform calculations with the gathered data to measure in-game metrics, which are then broadcasted to the server, to be stored and shown to the User.

3.2 Server

The Server is the main piece of the architecture, allowing the interaction between the User and the IMU Sensors.

The Server can broadcast instructions to the Raspberry Pi, controlling the IMU Sensors. When data is received from the Raspberry Pi's, the Server stores it, so that it can be shown to the User.

The Server also hosts the Client Application, where it receives the instructions to be sent to the Edge, and it shows the received data to the User.

3.3 Client

The Client Application provides the User an interface in which he can control the IMU Sensors, and displays the edge-computed in-game metrics.