UNIVERSIDAD DE COSTA RICA

SISTEMA DE ESTUDIOS DE POSGRADO

MULTIPLE MULTIPARTITE GRAPHS OCCLUSION SOLVER FOR SOCCER PLAYERS' TRACKING

Thesis proposal

LENNON NÚÑEZ MEOÑO

Ciudad Universitaria Rodrigo Facio, Costa Rica

2020

This thesis proposal was approved by the Assessor Committee, as a partial requirement to opt for the Academic Master's degree in Electrical Engineering.
Este anteproyecto fue aceptado por el Comité Asesor, como requisito parcial para optar por el grado de Maestría Académica en Ingeniería Eléctrica.

Dr. rer. nat. Francisco Siles Canales
Thesis Director

Ph.D. Esteban Meneses Rojas
Thesis Assessor

Ph.D. Marvin Coto Jiménez
Thesis Assessor

Lennon Núñez Meoño
Candidate

Contents

List of Tables

List of Figures

Glossary

Bad-Behaved Scene Frame from a soccer match video recording in which player blobs are agglomerated so that occlusion cases cannot be solved by the tracking algorithm.. vi, 32, 38, 39, 42, 45, 46

Dynamic Occlusion Event Occlusion event in which the occlusion's Agglomeration Size is variable and/or the event is comprised of smaller intermittent occlusion events, e.g. new players interacting with previously occluded ones at any point of the event, players constantly entering and leaving the occlusion event area, two players occluding intermittently. . 21

False-new-player Case in which a player's identity is lost during certain amount of frames, after which the same player is identified as a new one, therefore being assigned a new player identity label. E.g. have players identified as p1 and p2 during frames 40 to 45, during frames 46 to 50 p2 is not detected, but it is detected again as a player during frames 51 to 60. Due to being missing in frames 46 to 50, it is detected as a new player entering the scene, therefore, it is assigned label p3 instead of p2, with no connection in tracking history of labels p2 and p3.. 20

Identity Swap Case in which at least two players have their identity labels swapped. Usually occurs with players of the same team in which visual features and proximity lead the tracking algorithm to detect one as the other.. 6–8, 10, 11, 18, 20, 22

Medium-Well-Behaved Scene Frame from a soccer match video in which player blobs are somewhat spaced, tracking algorithm may or may not solve occlusion events.. 32, 38, 42

Occlusion Visual effect in which, due to the projection of a 3D scene into a 2D plane, the object of interest is visually blocked by another object from the same scene. Angle perspective of the camera and objects proximity affect occlusion perception. The occlusion is identified by its length (video frames) and event complexity.. 5–12, 14–16, 18–22, 37, 38

Occlusion Agglomeration Size Number of players involved in an occlusion event.. 21

Occlusion Dynamics Index Value that represents the complexity of the occlusion event's dynamics. It is calculated as the sum of the weights of different occlusion con. 21

Occlusion Event Complexity Score that depicts how complex an occlusion event is in terms of the involved objects interaction and the occlusion area objects density. The event can be categorized by its Dynamics, as Steady Occlusion Event or Dynamic Occlusion Event, and by its Agglomeration Size, represented by the number of players involved. It's the product of the Occlusion Dynamics Index and its Agglomeration Size.. 21, 38

Occlusion Event Region Region in the image plane where the occlusion event occurs. Its location changes accordingly to the players location, and its size may vary if it is a Dynamic Occlusion Event.. 22

Steady Occlusion Event Occlusion event in which the occlusion's Agglomeration Size is constant, i.e. the same occluded players are present from start to end of the event, and there is no intermittency. . 21

Well-Behaved Scene Frame from a soccer match video in which player blobs are spaced and may be present only simple occlusion events that the tracking algorithm can solve without needing an occlusion solver.. vi, 32, 38, 42, 45, 47

Acronyms

BPG Bipartite Graph. 9

DOE Dynamic Occlusion Event. 21, 22, *Glossary:* Dynamic Occlusion Event

MHT Multiple Hypothesis Tracking. 5, 10–13, 29, 30, 32, 34

MMPG Multiple Multipartite Graphs. 5, 12, 13, 16, 30, 32

MOT Multiple Object Tracking. 5, 7, 9, 19

MOTA Multiple Object Tracking Accuracy. 10

MPG Multipartite Graph. 5, 9, 12, 13, 25–30, 32

OAS Occlusion Agglomeration Size. 21, 22, *Glossary:* Occlusion Agglomeration Size

ODI Occlusion Dynamics Index. 21, 22, *Glossary:* Occlusion Dynamics Index

OEC Occlusion Event Complexity. 21, 22, 38, *Glossary:* Occlusion Event Complexity

OEC Occlusion Event Region. 22, *Glossary:* Occlusion Event Region

OECS Occlusion Event Complexity Score. 21–23

PDL Players Dispersion Level. 41, 42, 45

PF Particle Filter. 7, 8, 10, 11, 21, 25

ROI Region of Interest. 11, 21

SOE Steady Occlusion Event. 21, 22, *Glossary:* Steady Occlusion Event

Nomenclature

AgIx	Agglomeration Index. It has a value of 1 if there are only two players present in the occlusion event, and the number of players involved in any other case.
B	Number of branches from a branching node of a hypothesis tree.
b	Branch number of a hypothesis tree.
bn	Branching node of a hypothesis tree. It can also denote a branching layer.
CS	Complexity Score, calculated as $CS = DyIx * AgIx$.
DyIx	Dynamics Index.
G_i	Single graph, with position i inside a multipartite graph.
H	Number of hypothesis of an event.
h	Hypothesis number.
HT	Hypothesis tree.
k	Frame number.
k_f	Final frame.
k_i	Initial frame.
$mmpg^H_{k_i,k_f}$	Multiple multipartite graph starting on frame k_i and ending on frame k_f, of order H.
$mpg^{HT}_{b_1,b_2,...,b_i}$	Multipartite graph that is part of a hypothesis tree. It is located following each branch $(b_1, b_2, ..., b_b)$ from each respective branching node of a hypothesis tree.
mpg_{k_i,k_f}	Multipartite graph starting on frame k_i and ending on frame k_f.
N	Number of nodes.

n Node of a graph.

O Occlusion event.

o Occlusion event snippet.

$O_{k_i,k_f}^{complexity}$ Occlusion event from frame k_i to k_f, and complexity score *complexity*.

$o_{k,objects}$ Occlusion event snippet in the frame k, involving *objects* number of individuals.

w Window of frames used to perform object tracking in sections of the video.

w_K Window of size K.

w_{k_i,k_f} Window composed of the frames from k_o to k_f.

Examples:

- Let $mpg_{300,800,8}$ the 8th multipartite graph, of size g=501, result of the tracking process from frames k=300 to k=800 in the video window $w_{300,800}$.
- The graph G_4, of the multipartite graph $mpg_{8,501}$, has N=22 nodes, while the graph G_5 has N=18 nodes, there's certainly an error in the tracking.
- The $mmpg^1_{3,4,5}$ from the hypothesis tree HT=1 is located at the 3rd branch of the first branching node, 4th branch of the second branching and 5th branch of the third branching node.

Chapter 1

Introduction

This chapter covers the project's background and motivation, problem description and proposal of the solution to the problem stated.

1.1 Background and Motivation

Object tracking is used in different areas where it's important to know the motion of the objects of interest; some examples are submarine tracking, where a submarine needs to recognize projectiles and other ships moving alongside using passive sonars, aircraft surveillance with a similar scenario but using passive radars or an electronic warfare device to recognize the moving surroundings [49], in analysis of sports data [62] and, more recently, in medicine fields for cell tracking [36].

In terms of sport disciplines, the athletes usually use visual guides to improve and practice, be it recordings of the sport in execution or other visual aids on details of the discipline. The traditional method to analyze all the information available from those sources is to manually pick the scenes of interest and then select key plays to replay and study, the coach then will design a strategy based on the information extracted to improve the athlete's performance. Object tracking is introduced as a way to automatize data extraction for sports analysis, some disciplines require the tracking of an athlete's motion, in other cases like team sports, it is required to capture the motion of the many players involved. Players tracking goal is to provide information of the position of the player at any time during the whole match, by analyzing this simple piece of data it is possible to tell different statistics of its performance: total time played, distance traveled, area of the field covered during the match, speed, fatigue and placement of the player during different key plays [43]. Thus, players' performance and tactical evaluation can be assessed based on information from object tracking, making it a valuable tool for sports analysis.

One of the most popular sports in the world, with an estimate of more than 3.5 billion of fans, is the Association Football, or simply Football or Soccer [67]. It is a game in which two teams of 11 players, using any part of their bodies except their hands and arms, try to maneuver the ball into the opposing team's goal area, with a valid maneuvering field of about 90–120 m long and 45–90 m wide. The team that scores more goals wins at the end of a match [44].

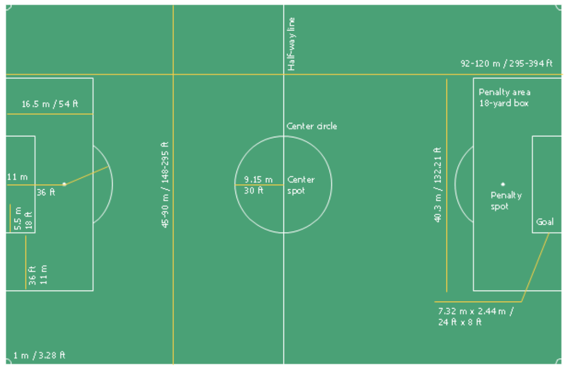


Figure 1.1: Football Soccer (Association Football) field. Source `https://www.conceptdraw.com/examples/football-pitch-dimensions-in-metres`

In the last years, some companies have dabbled into soccer players tracking for sports data analysis, the following are some examples of the most cited commercial applications in professional literature:

Kizanaro: It's a Uruguayan company started in 2008 as part of an academic project, with the purpose to provide services into soccer applied technology. It is used by the soccer teams of Angola, Uruguay, Paraguay, Venezuela, Colombia and others. The company states that it allows a fast visualization and analysis of the matches, obtaining information about the performance of the teams and players into video sections. The software classifies and breaks down the main plays of the match [27].

TRACAB: Optical Player Tracking solution from ChyronHego team, started in 2003 in Sweden. It's installed in about 300 soccer stadiums, being the official tracking technology used in leagues such as the English Premier League, German Bundesliga and Spanish La Liga, having also been selected for the major international UEFA and FIFA tournaments, as well as other sports than soccer. It uses camera arrays to record the field and delivers 3D information of the matches. The object tracking is supervised by a human operator (figure 1.2) [61].

Figure 1.2: TRACAB Soccer Players Tracking System: Camera array and supervised real time tracking. Source: `https://chyronhego.com/products/sports-tracking/tracab-optical-tracking/`

ChyronHego also offers others solution to sports analysis as an annotation tool, named Paint, used in sports news broadcasting to better visualize the different key plays. Nonetheless, the tool is a graphical visualization aid for sports journalists rather than a tracking or ”ground truth” generator framework [42].

STATS: Company with more than 35 years in the market working with sport fans and athletes. The company offers different solutions into sports analysis and fan engagement, for data capture the system SportVU is used, which offers performance statistics via extraction and processing of players and ball coordinates during the match, using High Definition (HD) cameras and statistical software and algorithms [58], [59].

Match Analysis: Started as a way to share the coach’s view to the team players, introduced into the market in 2000 in USA with data collection centrals in USA and Mexico. It brings a set of tools for video analysis and digital libraries to provide performance information and data storage, to perform tactical analysis of soccer teams. Real time data is collected and then synced with the video recordings by the analysts [33].

Since in 2015 the IFAB (International Football Association Board) allowed the use of techno-wearables during some soccer matches [14], many other technologies are also taking part into soccer player tracking, including different accessories into the players’ uniforms, one example is the use of GPS devices placed into the players’ shirts [21], [57]. However, optical players tracking has the advantage of not being intrusive on the players and that it requires just the video cameras and not any other support accessory, it’s cheaper and the methods can be exported to other areas on optical tracking.

Figure 1.3: Match Analysis system flow. Source: `http://matchanalysis.com/process.htm`

Due to the nature of private commercial software, it is not clear in general terms if the implementation is fully automated and which algorithms or methods are used to perform the object tracking tasks.

The Laboratory of Pattern Recognition and Intelligent Systems (PRIS-Lab) has developed a tool to perform Automated Soccer Players Tracking. This platform, named ACE, processes off-line the matches' video recordings, obtained from a single point of view, and returns the *tracklets* of the players' positions throughout the video. The current project bases its development on this platform, accessing the source code and soccer matches ground truth data base. The main goal is to improve robustness of the platform to occlusion events by implementing a Multiple Hypothesis method on it's multiple object tracking process.

1.2 Document Outline

The present work is organized as follows: first and current chapter is an introduction to object tracking and automatic soccer player tracking to define the field of research and the problem to solve, it develops the State of the Art i.e. the study of related professional literature to settle down the line of work, and the scope and objectives of the project; chapter 2 contains the theoretical background on optical object tracking and multiple hypothesis approaches; chapter 3 corresponds to the methodology, where framework and time frame are described; chapter 4 presents the preliminary results of the current ongoing implementation. The final section contains the appendices with different supporting documentation.

1.3 Professional Literature

There exist different methods and algorithms that have been developed to solve the many problems that arise when trying to track multiple objects, specially when these objects share so many features in common. M. Villalta [62], based on the Object Tracking survey by A. Yilmaz et al. [69], proposes a classification of these methods in three main classes according to the core functionality of the tracking algorithms, suited as Deterministic Methods, Probabilistic Methods and Data Association Methods. It is important to highlight that the order of appearance of these classes is a remark on the timeline of evolution of the approach mindset, as prior methods were not able to completely solve the Multiple Object Tracking (MOT) problem, it also depicts the focus transition when simpler issues were solved and the more complex had to be taken into account, like occlusionevents. Furthermore, M. Manafifard et al. [31] bring a detailed and extense survey on Player Tracking in soccer videos, reviewing about 163 different publications on the issue, studying cases like Multiple Hypothesis Tracking (MHT), Particle Filters, Graph Representation and Occlusion Detection and Handling. From this study it is important to highlight the following: detection (segmentation, background substraction) is critical in the performance of the tracker, occlusionsare the biggest challenge in MOT, MHT has improved different existent methods in occlusionsolving matters outperforming single hypothesis tracking, there is no mention of Multipartite Graph (MPG) approach as presented by [62] nor a mixture of MHT and MPG resulting in Multiple Multipartite Graphs (MMPG).

The following reviewed literature is used to illustrate the previous statements, the evolution of different approaches and to sustain MHT for occlusionmanaging. In addition, a subsection has been separated to discuss the works which implementations are focused on being robust to identity ambiguities and occlusionevents, appendix A depicts the distribution of reviewed methods in a table.

Deterministic Methods

These are methods that use visual characteristics to perform a deterministic trajectory tracking, i.e. there is a model that describes its motion and/or its appearance. The basis of most of these methods is to use brute force to find areas on an image that match a predefined and well known pattern. These methods fail as the track is lost when objects interact with others alike (players of the same team) and can't solve occlusionby itself.

The main approach in this category is known as template matching, where a predefined template is compared with different regions of the image, whenever it matches best, an object (player) would be identified [5], [50]. The problem with template matching is that the algorithm is fully dependent on its similarity with the objects of interest, if the template is not compatible with the players shape or color in a determined frame, it will fail. This is common on soccer player tracking due to the many visual changes a person has as it moves around the field turning around, crouching, falling and other shape shifting events, enforcing the necessity of a big and robust templates dataset to consider the many scenarios, also, the tracking result will have Identity Swap as the players of the same team have similar chromatic distribution.

Lefèvre introduces the use of active contours into soccer player tracking with a snake-based method [30]. The non-rigid nature of a soccer player is considered in this method, allowing a reconstruction of the contours of the players whenever its necessary, adding flexibility in comparison with a fixed template-matching. However, it would fail when occlusionhappens, as the contour may consider both players (in a 2 objects occlusioncase) as a single player, due to one player blocking the other in a 2D view.

To combat the occlusionproblem, some methods implement data fusion of information provided from different sources to aid the players detection. A template is updated with newer information from new observations over time, characteristics as size, texture and color are used to differentiate between players, and states are stored so that correct assignment is done after the occlusionpasses, which doesn't fully solve occlusionassignment. This method increases the complexity of the tracking platform as it incorporates an observation unit to update the templates, and also the template complexity by adding new features to recognize (color, size, texture). As it still depends on a template, the right update of the template and its initial state are crucial, which forces to manually declare the initial states of the match to aid tracking [35], anyways, similar and occluded players remains a problem.

Probabilistic Methods

Most of these methods are based on Bayesian inference and estimations to perform object tracking. They were introduced as a solution to optimal estimation of non-linear non-Gaussian state-space models, situations where the typical linear model-based Kalman Filter was not enough [13]. The main idea of these methods is to estimate the position of the object on a frame and update its likely position with new measurements, the measurements are assigned certain probability (weights) based on previous information until an optimal result is selected. It is important to highlight that these methods' decisions are based on probability, it chooses the most likely solution based on the cumulative probability of the assignments along the different frames, therefore, a wrong assignment will spread over time.

One of the most popular probabilistic algorithms for object tracking is known as Particle Filter (PF), introduced in 1993 as bootstrap filter, a numerical method for the solution of optimal sequential state estimation problems in non-linear non-Gaussian scenarios [38], and with increasing applications due to how the algorithm has evolved in hand with computational power. Due to its extended use in the topic it has been considered important to dedicate this section to its implementation.

Czyz, Ristic & Macq implement a color-based particle filtering algorithm, it was first designed to work with single object tracking, thus leading to wrap all players of the same team in a single global state. As particles are attracted to areas of high density, error from a single player is dragged through states update. The solution approached is to add a color histogram and to join detection and tracking tasks to distinguish players of the same team [11]. The experiments show that PF is a powerful tool on MOT, but the scenes analyzed did not consider occlusionevents, and the issue of particles moving towards areas of high density persists, which leads to Identity Swap of players that are close together. To solve the wrong assignment due to particle mixing, some authors have implemented single particle filtering for each player.

Single particle filters per player increases the computational processing requirements as the algorithm is applied for each player. Dearden, Demiris & Grau not only use Particle Filter for each player but decide to spatially segment the most likely areas where the player may move, thus including a spatial window and delimiting the possible solutions of the algorithm; PF is then applied on the delimited area [12]. This approach diminishes the influence of one player's error into the rest, however, the occlusionproblem remains. During an occlusion, the particles used may mix and get messy, therefore, the reliability of the data obtained throughout the event decreases the longer the players remain occluded. After the players separate from each other, using color-based recognition, the particles are reassigned to the respective player by estimating the trajectory that each player had before the event. Even though, the authors mention that particles mixing is still a possible outcome, so that all particles are assigned to a single player (therefore, losing the other player) or leading to Identity Swap with players of the same team.

Kristan et al. implement multiple single-trackers to confront the multi-target tracking problem, including closed-world assumptions to aid in the visuals processing [29]. The robustness of the algorithm is improved due to the constraints of a closed-world, as it was tested for indoor matches, however, those assumptions cannot be considered when working with outdoor sports, as outdoor conditions are not easy predictable (e.g. a cloud's shadow may cover the field totally or partially for a short or long period of time, it may rain or get foggy), also, the camera is statically placed above the court hanging from the ceiling, position that cannot be achieved in an outdoors stadium.

A major problem when dealing with Particle Filters is the degradation of particle assignment, specially when dealing with occlusion, as particles are wrongly assigned due to lose of information in the involved frames, which leads to losing players' identity after an occlusionevent (Identity Swap and new player condition after a split), therefore, PF alone cannot solve the problem. Itoh et al. [22] introduce a time-situation graph beforehand to an occlusionevent, the graph is used to solve the players' tracking lost after an occlusion, the quantity of players involved in the event is derived from the graph association prior and after the players interaction, thus allowing to assign the "new" players detected after the occlusionto the existing ones after the occlusion, fixing the splitting problem, but the swapping remains as the algorithm is based on PF. Similar approaches introduce the use of graphs to retain the states of the players so that they are not lost after an occlusionor in boundaries conditions as when player exits the frame (for broadcast recordings) or exits the field limits.

Data Association Methods

These methods address the object tracking as to find the optimal solution to a data association problem, where the data to associate is obtained from the object detection step. The main goal of these methods is to solve the wrong assignment and object track losing due to occlusionor similar players collision, most of the time used as a complement to a probabilistic/deterministic approach.

The simplest implementation of a data association method is using the Kalman filter due to its recursive nature, used to update the object-track assignments [47]. It is stated that for multiple-object tracking, there should be an additional computing to identify each detected object before actually tracking them, thus the method results are dependent on the right object detection step frame to frame.

The preferred representation of data for these methods is via a graph, were usually the nodes represent different states of a single player or team, and edges are weights that connect one state with the possible next. Soomro, K. et al. [55] implement a graph approach where each team is represented by a graph, in which the nodes correspond to player positions and the edge weights depend on spatial inter-player distance. They use color-based tracking to differentiate between teams to assign the player to one or another graph, and teams formations data to aid into the estimation of the position of the players, with the assumption that players remain in an organized manner, therefore, it is expected that error increases when players won't completely follow the initial formation. It also uses only video clips from broadcast matches with an overhead view, occlusionis dismissed.

Figueroa et al. use the graph representation for player tracking by focusing on the occlusionevents, where the nodes are used to represent different blobs to distinguish the players, aided with features as mentioned in above studies, like multi-camera view and color. Moreover, the study adds the use of morphological features for blob separation (like a pixeled player's size and shape model); the algorithm uses a backward and forward graph representation to optimize the graph and aid on multiple cameras to simplify the graph in occlusionevents. This is done by selecting the best view depending on the distance of the blob to the center of the field of camera view and which one brings the most information of isolation of a player, i.e. if in a view the occlusionoccurs, other camera can be used to verify if a player becomes isolated in that view [15], [16].

Wei-Lwun et al. use the graph representation to keep track of the previous assignments, using Bipartite Graphs (BPGs) to guarantee that one-to-one assignment is done by graph constrains; the graphs allow continuity from existing tracks, in contrast with other methods which nature tends to the creation of new tracks anytime an object is detected (as with particle filtering after occlusionwhen dealing with MOT). Wei's implementation considers the players as entities, each containing a set of features (faces, numbers on uniform, skin or hair color) that compose the Conditional Random Field (CRF), thus integrating different sources of information to improve the player recognition and tracking [66]. Siles, F. follows a similar approach by combining a set of features (color, texture, shape and kinematic model of soccer players) to the edges of a BPG to aid the tracklet-player assignment [52]. Later, Villalta, M. expanded the graph usage from a BPG to a MPG, the assignment is the solution of a graph energy minimization problem [62]. Villalta also includes a single panoramic view of the whole field by stitching the images from two 4K cameras used to record the matches. This approach eliminated problems related to resolution or temporal segmentation needed for clips from TV broadcast, however, occlusionproblems remain as the optical capture is from a single point of view, even though the use of MPGs improves the correct assignment of blobs identities, occlusionis not fully solved. It is also important to highlight that using two 4K images recording impacts in the amount of data to process (2x4k cameras of 3840x2160 px each, recording at 30 fps, leading to a load of 995 328 000 pixels per frame), thus, the use of high capacity computers like clusters or parallelization of the algorithm is needed to speed-up the process [63].

The typical implementation of a graph tracker requires to divide the whole video track into several blocks to process, where the initial state of the next block is defined by the graph solved from the previous one. There are two considerations on this: first, an error on any assignment (new blobs, Identity Swap, etc) will spread to the next block, second, the graph solving is a local solution for the block being contemplated, there exists information that's ignored on the current block because it is contained in further blocks, specially in occlusionmatters. Gedikli, S. et al. [19] implement a probabilistic method for estimating the trajectory of the players adding a MHT algorithm. The MHT preserves multiple hypotheses and their evolution through a defined temporal sequence (window), at the end of the window a decision is made with the best fitting hypothesis; Gedikli states that MHT was able solve some misclassifications result from occlusionevents that the typical tracker could not solve.

Multiple Hypothesis, initially presented as a solution for noisy and complex object tracking scenes in 1979 by Reid [60], is formerly widely used in scanner applications [2], where different measures are received on each scan cycle and spreading of a wrong assignment is critical in application such as projectile tracking, submarine location and others with a tight error acceptance. Moreover, it is recently extended to scenes considered complex due to the dynamic interaction between the objects to be tracked, like pedestrians and sport players tracking, where it outperforms single hypothesis algorithms [31], [70]. Due to its high computational cost, it is usually used as a complementary aid for other algorithms, commonly particle filters and/or a graph representation, instead of being used as the core algorithm of the tracker. Breitenstein's [7] experiments show that adding a MHT layer improves the results from a single algorithms implementation, in this case using MHT as an addition to PF, in particular in soccer with a Multiple Object Tracking Accuracy (MOTA) higher than 80% for the scenes selected for evaluation, however, identity switches persist due to the use of particle filters and robustness is low during the initial sequences.

Therefore, most improvements on tracking results due to robustness on occlusion events are found from a mixture of algorithms, usually by mixing MHT with other algorithms (Kalman filter, PF,) or similar approaches like a multi-layer outline, or, with PF based algorithms [4], [39], [71], where MHT has proven to be comparable to "stat-of-the-art methods on standard benchmark datasets" even in its early 90's implementation [26], [31].

Robustness against identity ambiguity

occlusionevent, as stated before with different examples, is one of the biggest problems in optical object tracking. Of the several ways to fight it, some examples are: improving the current algorithm with more features, omit occlusionand care only for previous and posterior identities, increase points of view adding more cameras, and treat the tracking as an optimization problem via multiple layers of assignment (typically via MHT or graphs plus additional criteria).

Hess & Fern improve the PF with their Discriminatively Trained Particle Filters [20], which is an alternative for the typical particle filtering approach by training the data set of the players with features as appearance (size, color), motion (via 11 different probabilistic features) and player interaction (to prevent particles mixing between different players), all features constantly updated frame to frame. This method increases the complexity of implementation of the algorithm as it requires to update the features and train the data set for each player, but shows great robustness when occlusionoccurs, especially considering that the algorithm was implemented in American Football, sport with more contact between players than Association Football.

Ok, Seo & Hong attack the particles mixing problem when using PFs by implementing an Occlusion Alarm Probability (OAP) [41], intended to repel the particles from sticking between players in multi-target tracking, thus preventing particles mixing and Identity Swap. The results show that there are some occlusioncases in which the repel leads to unassigned particles to one of the players, after the occlusionends, the player previously hidden is now considered as a new player (blob) entering the game due to the missing information from its previous states during the occlusion, this due to not considering tracking splits scenarios.

Sabirin et al. [46] detect early occlusionevents by defining a rectangular Region of Interest (ROI) to separate each player, rather than a tight contour, when two or more of these regions are overlapped by a 60% of their area, an occlusion is detected. In general, occluded players are considered a big single ROI during the occlusion event, their identities are assigned after the event, when there's enough information to identify the players during the occlusion, single regions are assigned preserving the shapes from previous frames, updated just after the event finishes. There's no complete identity solution during an occlusionevent, making the algorithm robust to occlusions (the identities are recovered after the event) but not occlusionsolving. It is important to highlight that Sabirin included players' texture as a feature for object identification, alongside other features as position and estimated motion based on optical flow in the hue channel, to identify different players, showing it as an useful feature allowing easily differentiation from different teams and also being the key on the algorithms robustness to occlusion. However, the method is dependent on the segmentation results, as many others, thus the importance of efforts on improving the frames segmentation.

Morais and Xu [37], [68], although working with the indoor sport futsal, both suggest the use of multiple cameras to improve the players distinction, as different planes will add position information that a single plane cannot easily obtain, the positions of the players are estimated using homography. Iwase, S. & Saito, H. implement multiple view images in outdoors scenarios locating cameras around the game field [23]. Even though the use of multiple cameras has proven to improve the segmentation results and diminish the occlusionissues (especially when the field is surrounded by cameras and 3D information is obtained), it is dependent on the optimal different views needed to prevent occlusion, which can anyway happen for non-rigid 3D objects in motion, e.g. one or more players may fall over a single player so that it is completely hidden from any camera, or there may be a group of more than two players colliding, which leads to players ambiguity associations. [28] also mixes information from multiple camera views into an MHT environment, using 3D information to aid correct identification of players during an occlusionevent. To prevent occlusionusing multiple cameras, new research questions may arise like: how many cameras? And, where to place them to cover the most, if not all, the 3D space? Thus, making the solution hardware dependant. In [31], the previous questions are exemplified by a variety of camera configurations, many different ways to place the camera and occlusion persists being a nuisance.

Jiang [24] represents the possible locations of a single object as nodes of a graph, building a multi-layer graph with the occupation probability of the object, a set of graphs would represent the multiple objects to track. Shitrit et al. [51], in a similar way, implement a graph representation for players' tracklets, reducing the problem to a multi-commodity network flow optimization, using K-shortest path to solve the assignments on a multi-layer graph. These representations work similar to a MHT approach, where different tracks are preserved and optimized to find the best later on, instead of picking local bests on each iteration of the algorithms. The above implementations resemblance a MPG with MHT approach, differing in the way the graph is constructed, which is considered on the current work's implementation as a MMPG.

In [32], it is used an appearance-based model MHT, in which they consider appearance and motion cues to refine players' identification. The combination of different features in order to improve the segmentation, as stated by Manafifard, is a needed modification to data association, as these methods usually start by considering that the nodes to associate (in a graph representation) are the right measurements. They show that an appearance-based model MHT performs better than a simple MHT algorithm, giving it the a better discrimination of players and false alarms. This consideration is taken into account in the current project, where the nodes of the graph are not only a label and a position, but a set of optical features that weight on the association solution phase.

MHT outperforms those methods using single hypothesis in the case of occlusion and noise [31], however, it is still not a flawless method, as it depends on the estimates of the measurements within an occluding blob, where a sudden change on velocity (reality differs highly from estimation) or a simple bad estimate results in tracking error [25]. Also, it has a high computational cost that makes it unattractive for real-time applications, nonetheless, it sets an interesting paradigm from which a multi-layer structure for assignment solving can be implemented [24], [51], which sets a different way to resolve the typical graph optimization problem. The previous leads to the idea of implementing a multi-hypothesis/multi-layer structure as a mixture of MHT and MPG, a Multiple Multipartite Graphs (MMPG) tracker.

1.4 Problem Description

Research questions

Will a Multiple Multipartite Graphs approach improve precision of soccer players' tracking in comparison to the current method based on single Multipartite Graphs?

Can a Multiple Hypothesis Tracking based Multiple Multipartite Graphs algorithm solve a higher percentage of simple occlusion events, as those that involve one player being occluded by another, in comparison to other state-of-the art algorithms?

Problem

Automatic multiple object tracking is not a flawless process, there are different events that make it a difficult task, in particular, the occlusion events. When performing optical object tracking, occlusion events cause information losing that leads to wrong or incomplete tracking results. Many algorithms have been implemented to solve this issue, but they fail one way or another, making it necessary to add a final layer of human aided correction, thus current tracking platforms are not fully automatic.

1.5 Hypothesis

By implementing a Multiple Multipartite Graphs data association tracking method, it is possible to improve the occlusion event management of the tracker, which will lead to a metric value of MOTA greater than 75% on a whole match.

1.6 Objectives

General Objective

Implement a Multiple Multipartite Graphs occlusion solver to manage occlusion events on automatic soccer players' tracking from Ultra High Definition video.

Specific Objectives

- Develop the framework needed for the Multiple Multipartite Graphs tracker.
- Design a metric to detect simple occlusion events on soccer matches.

- Implement Multiple Multipartite Graphs occlusion solver.
- Validate the algorithm against annotated ground truth datasets.
- Disseminate research to scientific community.

1.7 Scope

The current project focuses on the occlusion events problematic, it departs from the previous ACE platform version which includes multipartite graphs to perform players' tracking, using panoramic field view in ultra high definition (UHD) images from the stitching of two 4k recordings of each half of the soccer field. The platform will transform from using single multipartite graphs to a Multiple Hypothesis Tracking scheme, self-named Multiple Multipartite Graphs. Some other changes are considered to improve the platform, according to different specific needs of the project encountered during its implementation.

Chapter 2

Theoretical Background

This chapter covers important concepts and key theory regarding multiple object tracking, Occlusion, graph representation and the intuition on Multiple Multipartite Graphs (MMPG).

2.1 Multiple Object Tracking - Soccer Players Tracking

To perform soccer players tracking, a single player must be distinguishable from any other player along the match. This task, as easy as it may seem for the trained eye, is a hard task in object tracking, taking into account the next considerations: not relevant objects (e.g. goal posts, banners, audience) must be filtered from the objects of interest (players, ball, referee), each player must be identified in each single frame, the same player must be identified throughout the video (consecution of frames), the referee is not a player (can be or not removed from the *tracklets* depending on the desired analysis), there are active players substitutions during the match with fresh players, both teams fight for the same ball leading to occlusionand tons of players interaction, and many others. Therefore, object tracking is separated into different steps dedicated to specific tracking tasks.

Three key steps can be recognized: detection of objects of interest, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior. The whole tracking task can be defined as "the problem of estimating the trajectory of an object in the image plane as it moves around a scene" [69]. Tracking objects can be messy due to loss of information caused by projecting the 3D world onto a 2D image (some platforms like SportVU operate directly with 3D data from multiple views coupling), noise in images, resolution of the video recordings, complex object motion, scene illumination changes, partial and full object occlusions, and other problems application dependent; each step has its own difficulties.

The three key steps mentioned above can be recognized in the implementation of the computational platform ACE. The Laboratory of Patter Recognition and Intelligent Systems (PRIS-Lab) develops a computational platform for soccer digital video analysis, called ACE, platform with the task of providing automated soccer matches analysis to human motion scientist, coaches, teams and general public. This platform is implemented in a multi-layer architecturecomprised of two main stages: perception and cognitive [53].

First stage is covered by the temporal and spatial segmentation blocks. The temporal segmentation performs detection and classification of scenes with players information (for broadcasted soccer games) to remove those recording scenes that have no information of interest, like advertising spaces; if the soccer game is recorded in panoramic view of the whole field, this block is not needed. The spatial segmentation detects and localizes the objects of interest in each frame of the scene.

The second stage is covered by the trajectory tracking and semantics of the game's tracked activity. The trajectory tracking is where the position-object assignment is performed, using as input data the results from object segmentation. The semantic analysis performs a reinterpretation of the trajectories to detect actions and classify events, this block is responsible of generating statistics, occupational map and other information to provide to the customer [52].

Different methods and algorithms are used to try to solve each object tracking step. As the project focuses on the players trajectory tracking, only methods and algorithms directly related to the trajectory-object association will be discussed. The inputs for the algorithms to study are the objects spatially segmented by the previous block in ACE's flow, i.e. **player blobs**, block which graphically identifies them drawing a contour around the player's detected shape, the centroid of the contour is used as the estimated position of the player. Therefore, it is important to highlight the influence of the results of the segmentation task into the trajectory tracking. Details of the base implementation of ACE platform are discussed in [62].

To understand the complexity of assigning the positions detected in a set of frames to a player and identify its trajectory, it is important to know how a typical soccer game is broken down. First, the game officially starts with all 22 players ”easily” distinguished as they are separated one another following a game strategy formation, and both teams are in opposite sides of the field. As the game begins, the players start to move “randomly” over the field, a player may decide to stay in position, move arms or crouch (shape change), jump, go forward or backward, left or right; as the ball moves from one side of the field to the other, the player may change its previous intended trajectory to another; the motion is errant and not much predictable, even though there is a general strategic plan. If the players would not interact one with another, the trajectory-player assignment may not be as hard, but, as the game states, both teams are fighting for the same ball, this leads to more than one player sharing a close range to the ball as they struggle to control it, possibly generating occlusionwhen one player is placed behind another (from the perspective of the 2D image recorded from the same view plane); which player is which? Using the colors of the soccer suits is useful only if the players are of different teams, but if the players are of the same it won't help much, even less if it is confused with the background (see Figure 2.1).

Figure 2.1: Players detection issues due to visual properties, scene conditions and segmentation process. Results from ACE platform, 2018

Automatic object tracking has to deal with previous mentioned issues and many others. Most of these issues correspond to the first stage, specifically from spatial segmentation, as they are dependent on the image characteristics. A perfect segmentation or feature extraction would simplify the second stage, where the association algorithms could operate with clean and organized information. However, as these issues are embedded in the image, data association and object identification has to deal with messy and flawed detected objects. Even though there are different methods that try to improve the quality of the detection phase results, some events persist to the other phases, like occlusion events. Therefore, it is required that the identity assignment stage takes charge of those problems, resulting in the previously mentioned errors like Identity Swap, lost and spurious objects.

2.2 Occlusion

When using images, the features extracted depict visual characteristics of the scene under evaluation, one way or another, they describe each object contained in the scene. However, images are but a projection of a 3D world into a 2D plane, where these visual characteristics are transformed and even hidden, making them hard to extract. Typical features extracted in this context relay on pixel color (using histograms of color in different representations like YUV, RGB or HSV) to detect areas of interest in the image (background extraction, possible objects detection) and shape/size of the objects to filter spurious detections. However, some information is lost in the 3D to 2D world transformation, sometimes vital like depth location of the objects, which leads to occlusion events.

In an occlusionevent, objects that share a common axis location, but different depth from the camera, appear to share the same 3D space, as depth information is lost in the plane of the image. When this occurs, it is hard to identify which object is which, as figures and colors are mixed and the common segmentation methods fail to distinguish each object involved, in most cases; this particular scenario, as stated in [31], is the biggest challenge in Multiple Object Tracking (MOT). Figure 2.2 illustrates occlusion events on real soccer matches, even though two close players are not visually blocked for the human eye, its proximity and visual features can lead either segmentation or tracking algorithms to give a faulty result (figure 2.2a). Some other events are more complex and practically impossible to automatically correct (figure 2.2b).

(a) Minimum proximity between players is considered as an occlusion event by the algorithm, which groups both players as a single one. [46]

(b) Four players occlusion on TV broadcast video.

Figure 2.2: Examples of scenarios perceived as occlusion events.

The existence of occlusion events, and a segmentation process that is easily affected by it and many other noisy situations, makes it a need for the identity assignment phase to be robust in order to deliver a clean tracking output, it must deal with false detections and missing information. However, such a perfectly robust algorithm does not currently exist, and the current ones which outputs have the higher accuracy percentages are not fully reliable on its own. Recent approaches tend to mix different methods to aid the core algorithm into solving assignments, be it hardware solutions like increasing the points of view, or a mixture of feature models and tools [31]. In terms of occlusion, there are mainly three ways to *confront* the event: no-handling, event detection and event solving, where the first two are the most common.

No-handling implies that occlusionis not considered as an input event for the identity assignment. These methods usually rely on the ability of the algorithm to make corrections or its robustness to false detections. As discussed previously, there is no flawless algorithm, and this approach will lead to an incomplete tracking history that requires human aided cleaning of the results. Other way of not handling the occlusionis trying to prevent it from happening on the scene, this is done by increasing the points of view to perform a 3D projection of the field. With depth information, most occlusionscenarios are dismissed in exchange of having to care about homography and camera placement, some other occlusion events are still present (like in figure 2.2b) and the algorithm is fully dependent on the segmentation results.

Event detection is the ability to recognize an occlusionevent, usually indirectly by human interpretation of tracking information, but to dismiss it, i.e. identities are assigned before and after the frames in which the occlusionoccurs, but during the event there is no verification of the identities. This can be seen in [15], [16], where the missing nodes in the sequential graph hints an occlusionevent (the graph passes from having two identified players to only one during many frames, and then again having two players, see figure 2.3). However, there is no occlusionhandling, the missing information is not corrected and the tracking output only shows the player missing in some frames, but it is considered enough to know the players' position before and after the occlusion, as sometimes it takes few frames, in the best of cases. Usually, scenes are not so simple and there occurs identity swappingand false-new-playerlabeling after the occlusionevent.

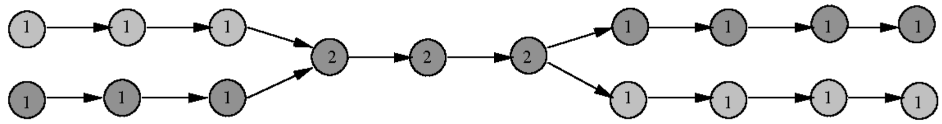


Figure 2.3: Sequential graph where missing identities suggest an occlusion event [16]

Some occlusiondetection techniques are used to improve the main algorithm behaviour, like in [41] where occlusionsare detected by a proximity alarm in order to disperse the particles from the Particle Filter (PF) that is used to track players. Another example is [46], that uses occlusiondetection to trigger occlusionhandling.

occlusionhandling refers to trying to solve identity assignment during an occlusionevent, and/or to verify and correct identity assignment before and after the occlusionoccurs. Some examples are [15], that builds forward and backward graphs of the frames with missing information in order to merge data prior and after the occlusionoccurs; [46] uses different player attributes (like color and movement direction) and prior-occlusion Region of Interest (ROI) placement to define the position of missing players during an occlusionevent.

Handling and solving occlusionrequires at least three phases: detection of the event (in which frame it starts and when it ends), revealing hidden players and identity preservation during the event. It is important to note that this is a simplification of what it would take to solve occlusion, as there are different degrees of complexity in which it can occur depending on the amount of players involved and if there are other dynamics like players intermittently occluded while others are permanently occluded.

Occlusion event classification and identification

Occlusion events are different and vary in complexity. Simple events as a short sequence of two players can be solved in most cases, for example with a forward-backward sequential graph [15] or with PF [20], however, even in these scenarios occlusion solving effectivity is not 100% and there exists many other scenarios more complex that are yet to be confronted.

An occlusion event O can be identified by its length, in video frames, and its complexity. The length can be represented either by the number of frames or by the initial and final frames k_i, k_f. The Occlusion Event Complexity (OEC) is a score that depicts how complex an occlusion event is in terms of the involved objects interaction and the occlusion area objects density, where $O_{5,30,1}$ is an occlusionevent that occurs from frame $k = 5$ to $k = 30$ and has an Occlusion Event Complexity Score (OECS) $CS = 1$, where the OECS is the product of the Occlusion Dynamics Index (ODI) and its Occlusion Agglomeration Size (OAS), to be detailed later.

To study the event's complexity, it can be categorized by its Dynamics, as Steady Occlusion Event (SOE) or Dynamic Occlusion Event (DOE), and by its Agglomeration Size, represented by the number of players involved. From the dynamics perspective, it considers the interactions of the players during the event. It may happen that during the whole occlusion there were only two players present, both move to posses the ball, fight for it for some frames, and then one takes possession of the ball and runs faster than the other bringing an end to the occlusion. The previous example depicts an event in which the OAS is constant, but it may vary in more complex events.

In the case of an occlusionthat starts with two players for a couple of frames and later changes to three players when a new one gets close to aid its teammate, the OAS varies, and thus the complexity of the scene increases. In this case, let $O_{5,30,OAS}$ be the occlusionevent described previously, and $o_{5,10,2}$ and $o_{11,30,3}$ be the sub-events that comprise $O_{5,30,OAS}$, it can be noted that the OAS is useful to break off a complex event into several simpler ones. However, the event can get even more complex by simply adding intermittency to one of the players.

If we have the previous example but addition intermittency, let's say that the third player enters and leaves the Occlusion Event Region (OER) several times, the OEC increases as the tracking algorithm must deal with an intermittent merge/split for several frames, result of the third player interacting with the previously occluded pair. If there are many intermittent players, the complexity of the scene increases even more. Thus, we identify a SOE as one that has no intermittency, with an ODI weight of 0, and a DOE as one with intermittency, with an ODI weight of the number of intermittent players.

Other factor to consider for the OEC is if the players interacting are from the same team, as similar colour distribution leads to Identity Swap, it is easier to distinguish players from different teams. In this sense, we define an ODI weight of 1 if there are players from different teams, an a weight of 2 otherwise. The final ODI is the sumatory of the intermittency weight and the team weight, e.g. an occlusionevent with two players of the same team (team weight of 2) where one is intermittent (intermittency weight of 1) would have an ODI of $DyIx = 3$. To calculate the final OECS, we use its OAS, with a value of $AgIx = 1$, as there are only two players present, and multiply it by the $DyIx$, therefore, the OECS for the *simple* event from the previous example would be $CS = DyIx * AgIx = 3 * 1 = 3$, while an scenario with the same OAS but with no intermittency and players from different teams ($DyIx = 1$) would end up with an OECS of $CS = 1$, which is the simplest possible occlusionevent (a $CS = 0$ would mean no occlusion is happening).

This project will focus on occlusion events with an OECS $CS = 1, 2$, which are simple occlusion events, cases where $CS >= 3$ will be tested, as those are interesting and challenging scenarios (specially if the high score is due to intermittency), but no specific result is pursued over it. The most common

OECS are yet to be obtained from scene labeling on real matches, to be shown in further results.

2.3 The Graph Representation

A *graph* G is a mathematical structure that shows relations between the objects it comprises, it mainly consists of two things, a set $V = V(G)$ whose elements are called *vertices, points,* or *nodes* of G, and a set $E = E(G)$ of unordered pairs of distinct vertices called *edges* of G, such a graph is denoted by $G(V, E)$. If there is an edge $e = u, v$, i.e. the edge e connects vertices u and v, then vertices u and v are *adjacent* and are the *endpoints* of e [45].

To easily understand these relations, graphs are pictured by diagrams where each vertex v in V is represented by a dot or a small circle, and each edge $e = v_1, v_2$ is represented by a curve which connects its endpoints v_1 and v_2 (see figure 2.4 for a generic example of a graph).

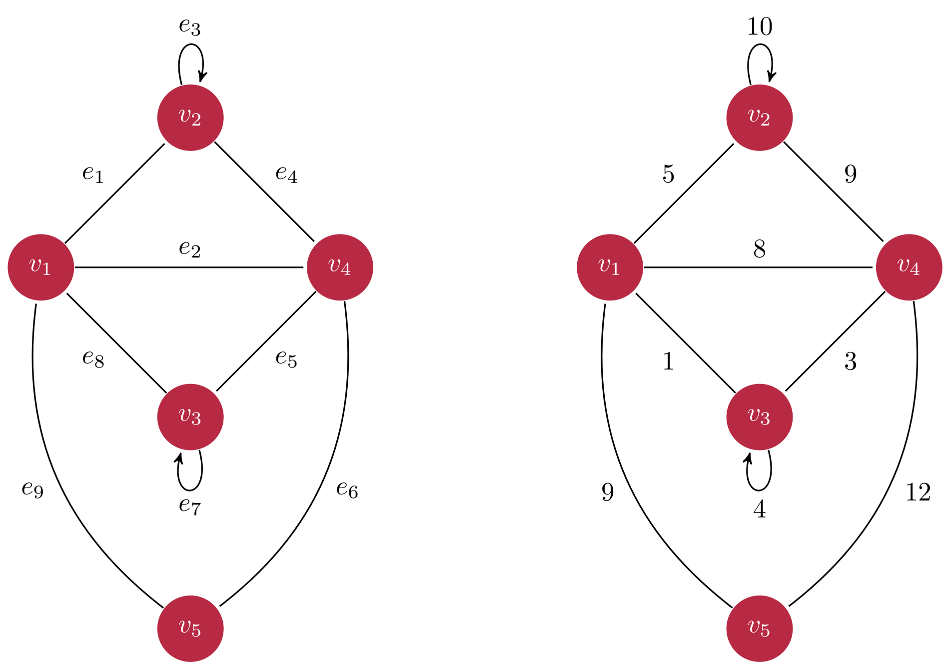


Figure 2.4: Graph G pictured by its vertices and edges relations (Left), example of a weighted graph (Right).

If graph G is assigned data to its edges and/or vertices, it is called a *labeled* graph, in particular, G is called a *weighted* graph if each edge e is assigned a number $w(e)$ called the *weight* or *length* of e, and thus an *edge-weighted* graph (see figure 2.4).

A *complete* graph is a graph G in which every vertex is connected to every other vertex in G (see figure 2.5). A graph is *directed* if it is made up of a set of vertices connected by edges, where the edges have a direction associated with them.

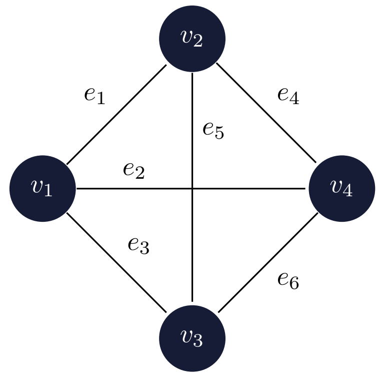


Figure 2.5: Complete graph example

A *multipartite* or *k-partite* graph G is one in which its vertices V can be partitioned into k different independent subsets such that vertices of the same subset are not adjacent. If the amount of subsets $k = 2$, it is said to be a *bipartite* graph, if $k \geq 3$, it is said to be a *k-partite* graph. It is a *complete k-partite* graph if every pair of graph vertices in the k sets are adjacent, if the subsets have $p, q, ..., r$ vertices respectively, the complete k-partite graph is denoted $K_{p,q,...,r}$ (see figure 2.6).

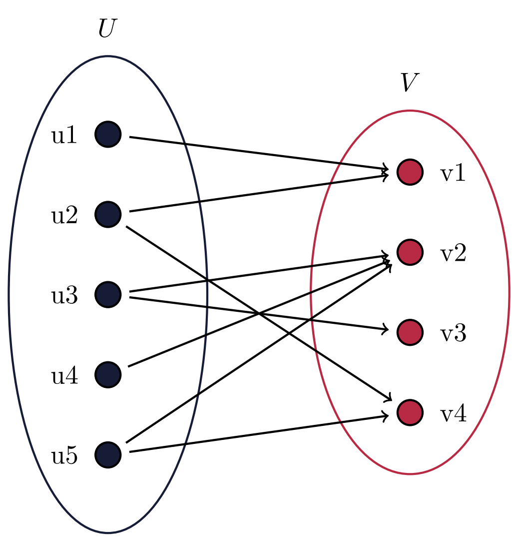


Figure 2.6: Bipartite Graph Example

In the sense of ACE platform, a series of k frames are represented by a *directed edge-weighted k-partite* graph *mpg*, where the vertices (nodes) of each subset k represent a set of visual characteristics for each object of interest (players, ball and umpire), and the weighted edges represent a cost of making the nodes adjacent (i.e. the weight of node u_i from subset k-1 representing the same object as node v_j

from subset k). The *mpg* is initiated as a *complete* graph, and is then minimized by a shortest-path algorithm to obtain a graph where **ideally** each node has exactly one adjacency (see figure 2.7). If there were no faults when tracking the players, each subset k would have the same exact number of nodes and the ideal minimized graph would be the common output, however, that is not the case in a real scenario.

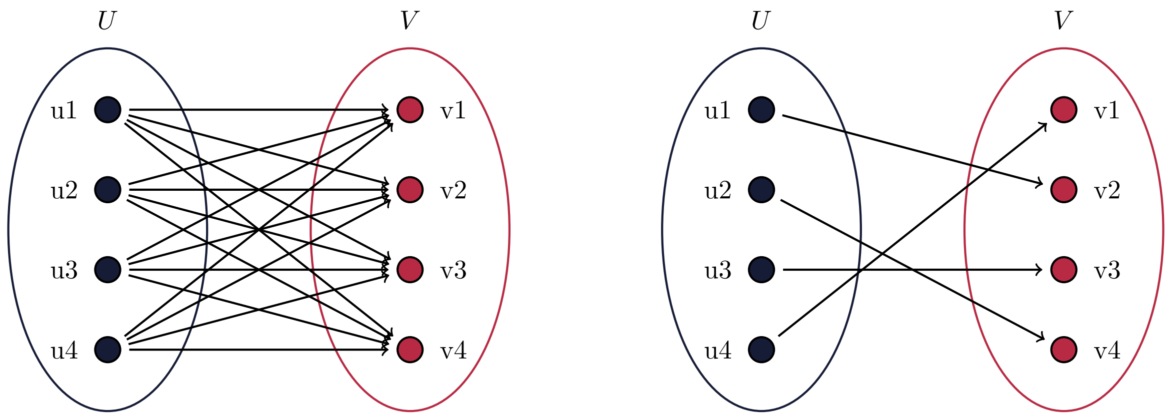


Figure 2.7: Bipartite graph example with 4 nodes in each subset. Left: Complete Ideal Bipartite Graph. Right: Minimized Ideal Bipartite Graph.

2.4 Multiple Multipartite Graphs

Object tracking involves detecting and identifying the object of interest in an image k, and preserving its identity in the next images $\{k+1, ..., k+i\}$ from a sequence of consecutive images (detecting and identifying the same object as itself in each image). Different features are extracted to estimate the object's position in the image, in this sense, the identity assignation and position estimation are considered to be an *assignment hypothesis*. Different considerations are taken to determine if that hypothesis actually represents the desired object, in a probabilistic tracking approach, each possible assignment with a high probability is a hypothesis (for example each particle in a PF[13]), where the tracking result can be a single hypothesis with the highest probability, the average of the hypotheses with highest probability, or a region of high probability. In a data association tracking method, in particular a Multipartite Graph (MPG) approach, each node adjacency combination is a hypothesis (see figure 2.8), it can be said that a complete MPG is one that represents the whole hypothesis space.

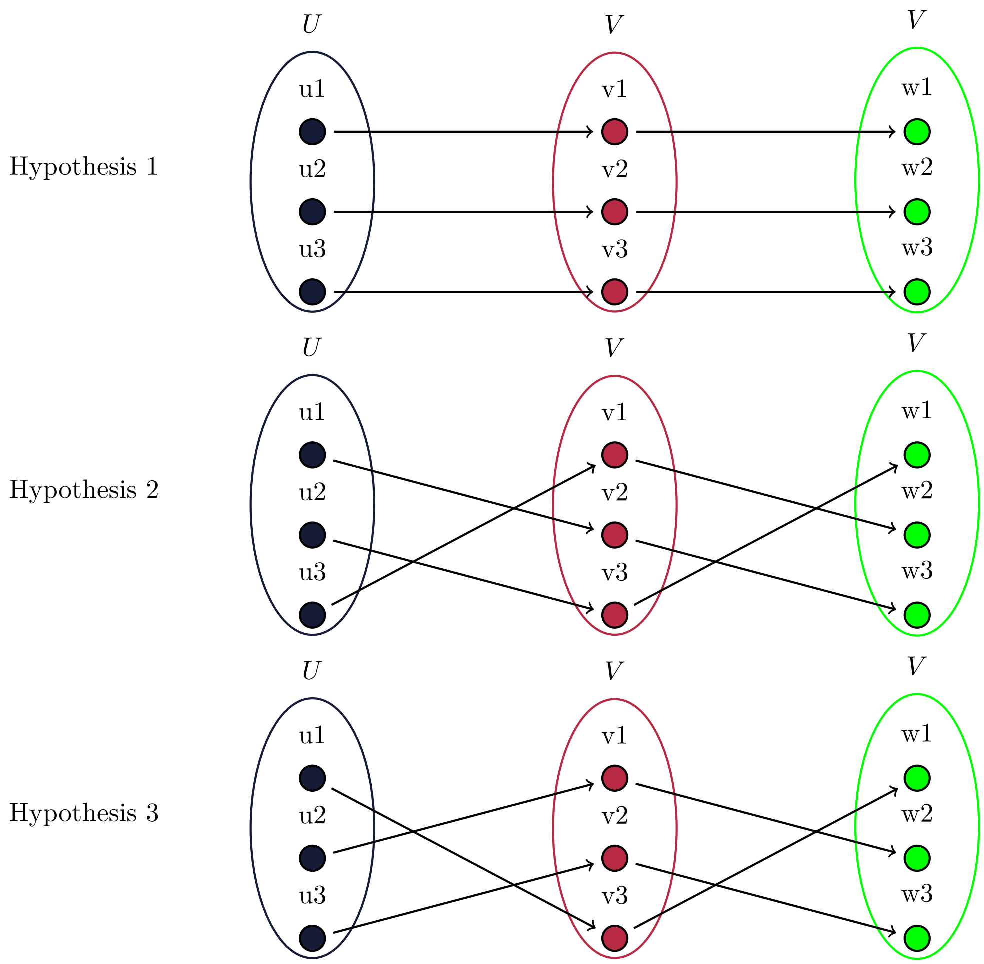


Figure 2.8: 3 assignation hypothesis examples for a multipartite graph of 3 subsets with 3 nodes each.

When considered the features extracted and other constraints (like distance and speed constraints) many of those hypotheses can be easily discarded, as they are *not possible* due to known physical interactions (a player cannot be in one side of the field on frame k and in the other side of the filed on frame $k+1$, if consecutive frames represent a lapse less than a second). For the *plausible* hypotheses, usually a shortest-path algorithm is used to determine the best fit minimized graph according to the different weights assigned to each edge (from features and constraints). However, the usual approach is to build a MPG for a fixed window size of K frames, and pick the very *best* assignment hypothesis and discard the rest, which would be the final solution if we suppose that the data from which the MPG was built is accurate with no noisy, spurious or missing identities, what happens in real case scenarios, thus the local best hypothesis may not be the global best, or even the best for a MPG that considers the frames from the same window and the next one.

One way to illustrate it is by comparing the hypothesis decision by shortest-path to what sometimes occurs with a GPS map navigator. It can occur that the driver is suggested to take an exit to go through a path which is just 1 second faster than the current course, the driver may decide not to do so considering that 1 second is not gain enough and that taking the exit would lead to a known hill were the car will be slowed, therefore lasting even more than estimated, the algorithm is designed to pick the *shortest* path from the currently available data. It can be argued that incorporating a way to calculate the estimated time of arrival (ETA) considering the hill slow down would improve ETA's accuracy, then, let's consider the case in which the navigator does not know beforehand every route from a map, but it generates the best route evaluating a portion of the map by fixed size windows, then, until the window with the portion of the map that includes the hill is processed the algorithm won't take into account its existence and slow down, therefore, it may still offer the driver to take the next exit for a shorter path, but just after that path was considered to be the fastest, the algorithm notices the hill that makes the current path not a global optimum, but a local best. That's the scenario when tracking football players (specially if the final goal is to track live players), while the current window is being processed, the algorithm has no idea of what the players will do (the possible routes), thus, the shortest-path algorithm picks the local best hypothesis, which may not coincide with a global best hypothesis.

The above scenario may have a real effect in motion flow cuts, for example a motion scene being composed of two MPG due to window sizing and sequential graphs creation instead of a single MPG from the beginning, like an occlusion event split in several MPG; in the end, the final graph representing the whole match is a chain of local bests hypothesis added sequentially, which may not have the same results as a MPG made from the entirety of the match's frames, however, more study is required to fully confirm it for different window sizes. Figures 2.9 and 2.10 compares the general processing of the MPGs of a scene comprised of 6 frames with a window size 3 and 6.

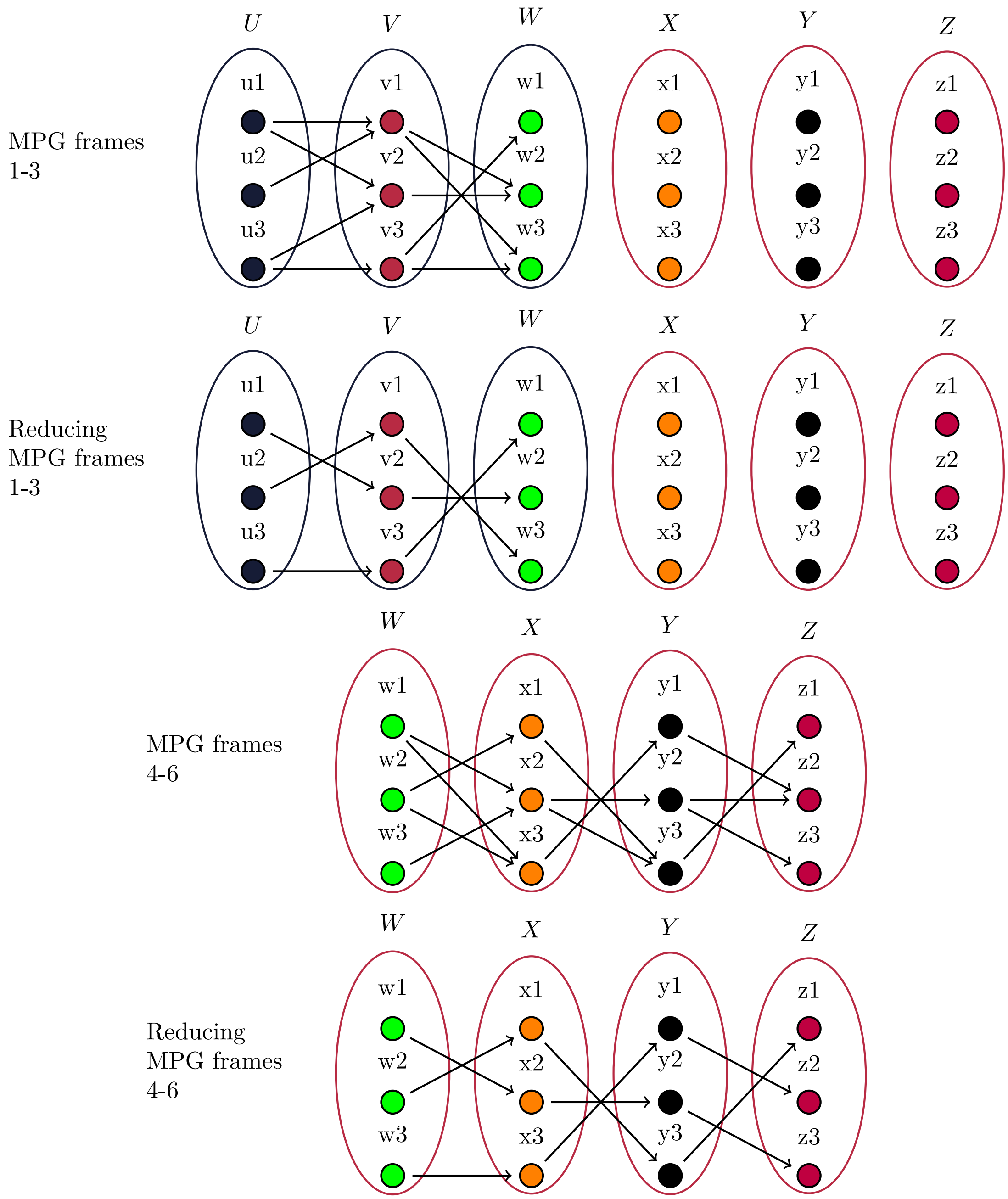


Figure 2.9: Multipartite graph processing with a window size 3 of a scene comprised of 6 frames. Subsets U, V and W represent the first 3 frames of the scene, as the window size is 3, these subsets comprise the first MPG, it is minimized and the final subset W is the starting point for the minimization of the next MPG for the last 3 frames of the scene.

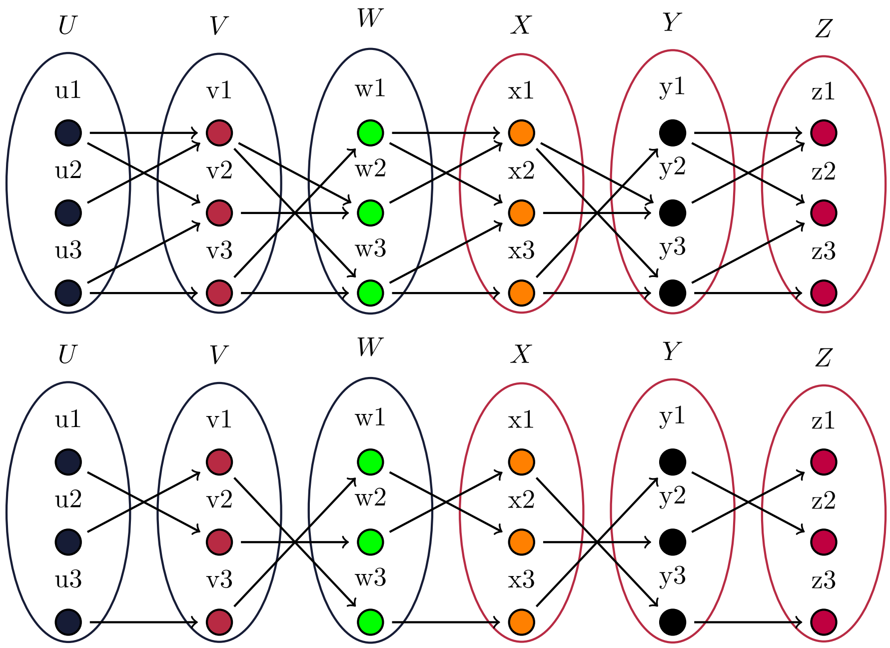


Figure 2.10: Multipartite graph processing with a window size 6 of a scene comprised of 6 frames. Subsets U, V, W, X, Y and Z represent the whole scene in a single MPG, which is then minimized with the whole scene's information.

Supposing that tracking can be improved by holding the hypothesis selection until there is more information for an actual best pick (we have knowledge that there is a hill coming, so let's be flexible with the MPGs), MPGs of different sizes would be created to encompass complete motion scenes, i.e. to isolate occlusion scenes (the hill) on their own window separate from the other frames, keeping hypotheses from previous windows. The idea of propagating multiple assignments hypothesis was first presented in 1974 by Singer, Sea and Housewright [54] to be used in dense multitarget environments, and later expanded in 1979 by Reid [60], presented as Multiple Hypothesis Tracking (MHT) algorithm, a systematic approach to solve tracking of multiple objects in a complex scenarios. The algorithm mainly consists on creating assignation hypothesis and propagating them as new measurements and detections are incorporated into the assignment space, usually depicted as a decision tree that encompasses the hypothesis space (figure 2.11). Those discarded hypothesis (branches) are *pruned* from the *hypothesis tree*, each new measure adds new information and thus creates a new subset of hypotheses (*hypothesis branches*).

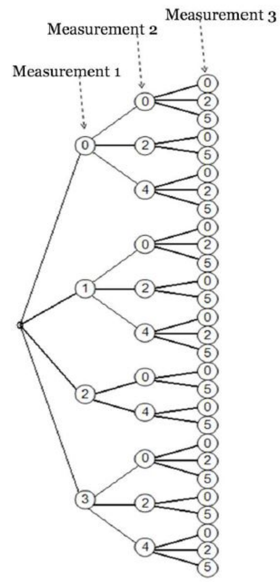
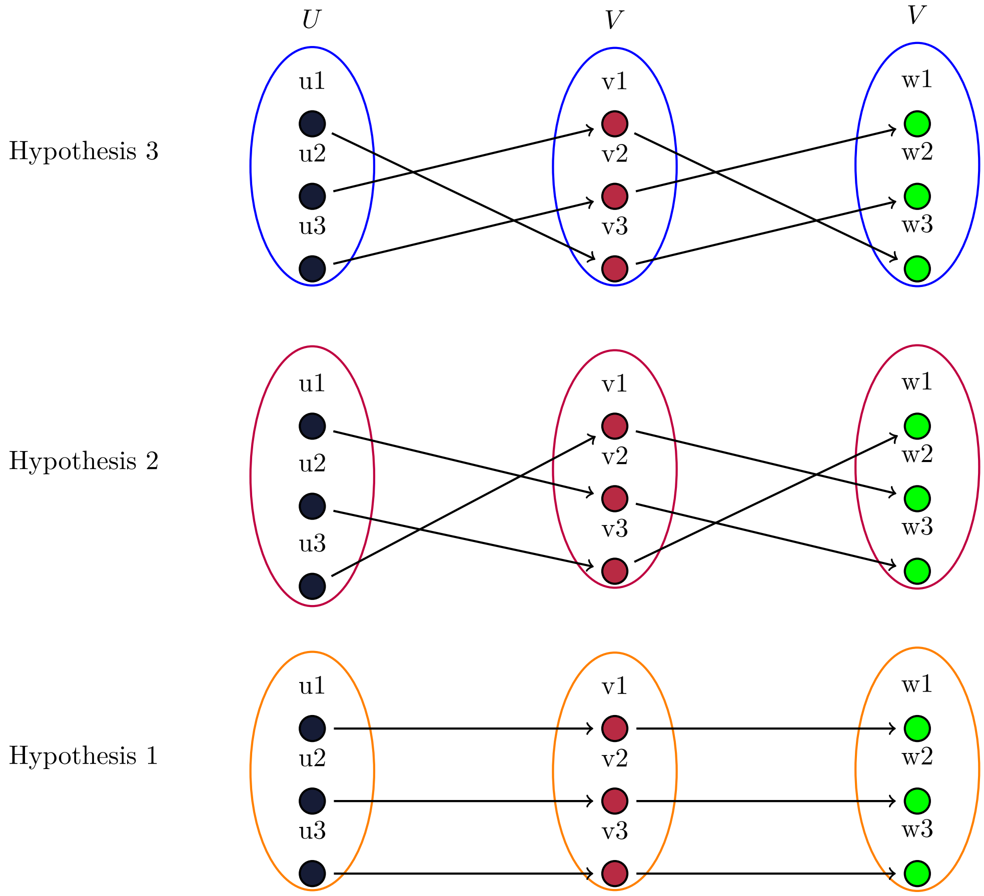
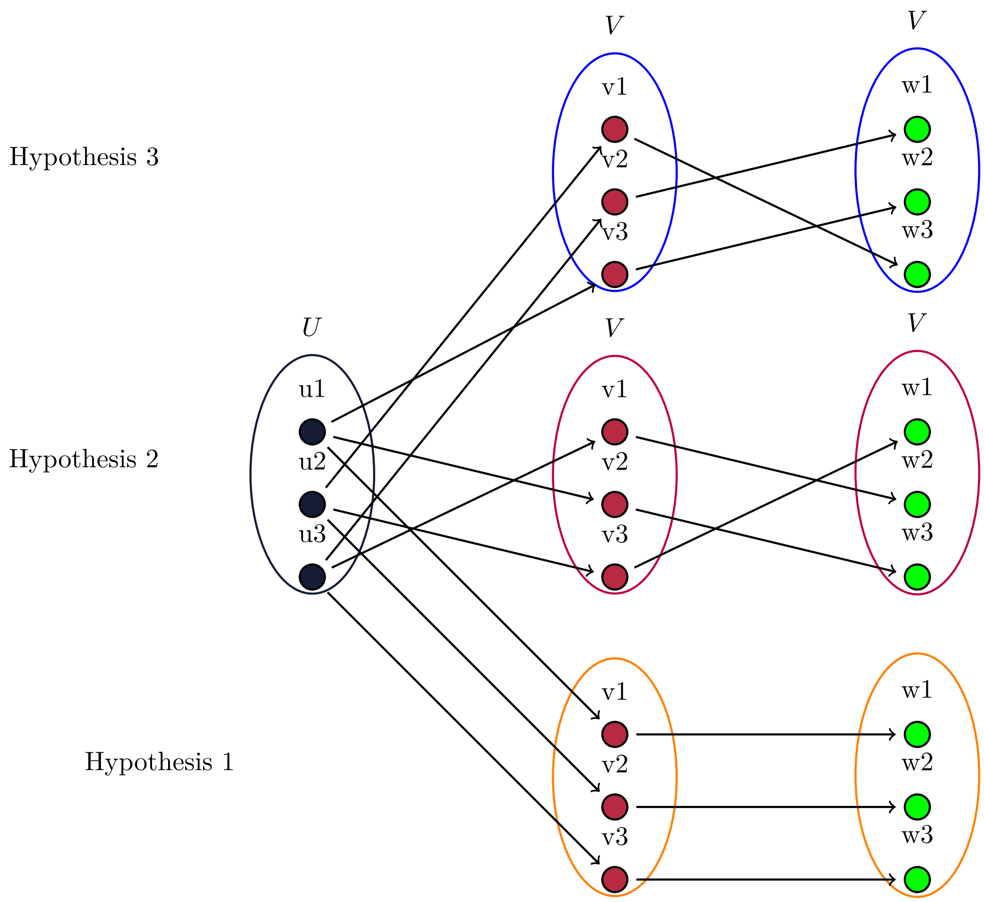


Figure 2.11: Hypothesis Tree: tree representation of formed hypotheses. [2]

There exist different implementations of a MHT algorithm [2], [3], [6], [10], [18], [60], as well as variations depending on the area of research, improvements or the algorithm it is mixed with [7], [19], [28], [32], [39], [71], but two key aspect remains, the first one is its high complexity and computational cost due to the hypothesis creation process (hypothesis can become virtually infinite if not limited) that makes it a requirement to use a robust pruning method and to limit its length with a fixed maximum size. This issue is similar to the one previously discussed with MPG, however, due to its complexity and cost, it is not desired to use MHT for the entirety of the match but rather in sections where it may come handy, like in occlusion events (here it is paramount the idea of occlusion events detection and classification). The second key concept regarding MHT is its tree-like shape, comparing the main structure of a *Hypothesis Oriented* hypothesis tree [2] with a MPG (figures 2.8 and 2.10), where each branch of the tree is a hypothesis and each possible assignment combination in a MPG is a hypothesis (taking each subset as a single node of a hypothesis branch), the possible MPGs can be arranged each as a branch of a hypothesis tree, thus making the whole hypothesis tree into a MMPG space (figure 2.12, where *hypothesis pruning* would mean to discard a MPG, and *hypothesis propagation* the incorporation of new subsets, i.e. expanding each hypothesis/MPG with new *measurements* from the following video frames. Each new subsequent MPG window adds a new set of *hypothesis branches* to the existent ones, making the *hypothesis tree* grow.


U
V
V
Hypothesis 3
u1
u2
u3
v1
v2
v3
w1
w2
w3
Hypothesis 2
Hypothesis 1


V
V
Hypothesis 3
U
Hypothesis 2
Hypothesis 1
u1
u2
u3
v1
v2
v3
w1
w2
w3

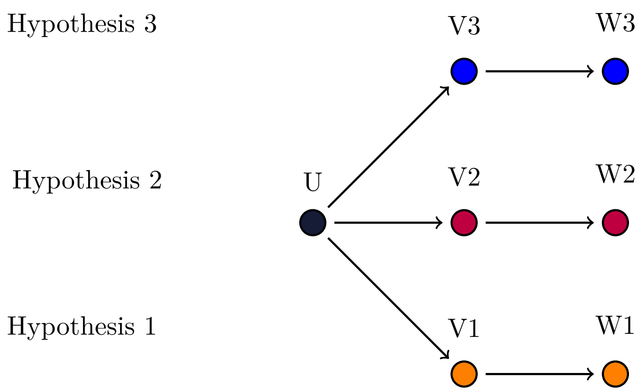


Figure 2.12: Depiction of Multiple Multipartite Graphs structure based on a Hypothesis Tree, where the first common subset is the starting point, each branch is a hypothesis composed of a possible reduced MPG

Therefore, soccer players' tracking would be composed of MPG, for Well-Behavedscenes, and MMPG for Medium-Well-Behavedand Bad-Behavedscenes, one solved by a typical shortest-path solver, and the others with a MHT-like approach.

Chapter 3

Methodology

The following tasks and considerations are required to achieve completeness of the projects' objectives, separated into three phases: Structure of theoretical background and state of the art, Implementation of occlusion handling into ACE platform, and Validation of tracking tool.

1. Structure of theoretical background and state of the art

    - Review professional literature related to multiple object tracking, sports analysis and state of the art methods and algorithms on the subject. The literature to be reviewed is maintained in trusted repositories, as IEEE Explore®, ACM®, ScienceDirect® and Springer®.
    - Structure reviewed literature according to distinct approaches on multiple object tracking and occlusion detection and handling.
    - The algorithm to be implemented is an improvement on the current development of the automatic players' tracking platform ACE, therefore, this platform functionality and structure will be studied in order to include the multiple hypothesis layer.
    - Define occlusion matters: occlusion detection, occlusion identification, occlusion handling, occlusion measure system. Perform a thorough study of how occlusion occurs and its effects on optical tracking systems, specifically on soccer matches.

2. Implementation of occlusion handling into ACE platform

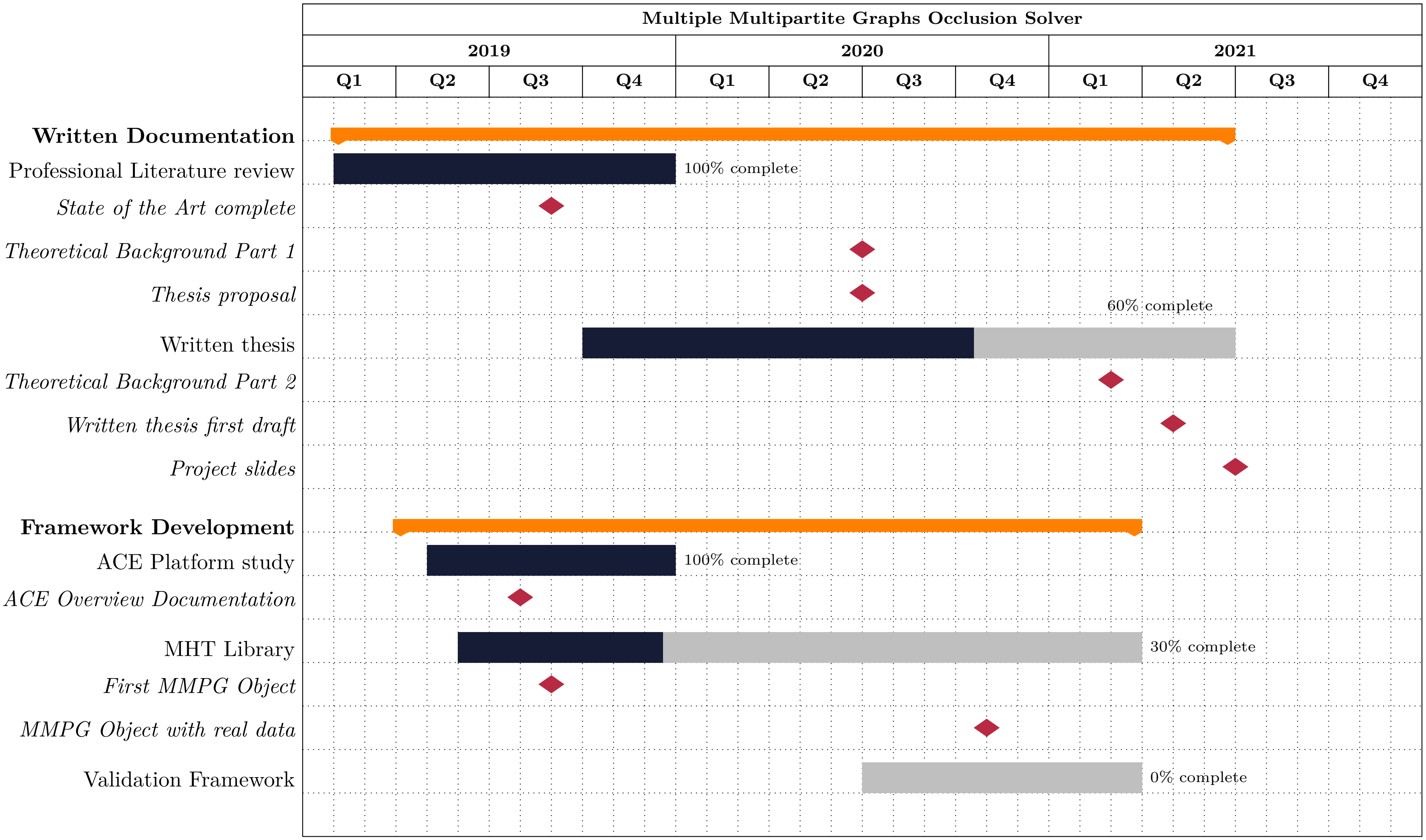
    - The platform to be used is composed of already existing libraries written in C++, from the platform ACE and other trusted open-source developers as OpenCV, as well as libraries to be developed according to the needs of the project.
    - Multiple Hypothesis Tracking algorithm implementation is to be studied from professional literature that depicts its implementation and special considerations, as well as from different already implemented free access libraries, to develop a custom library.

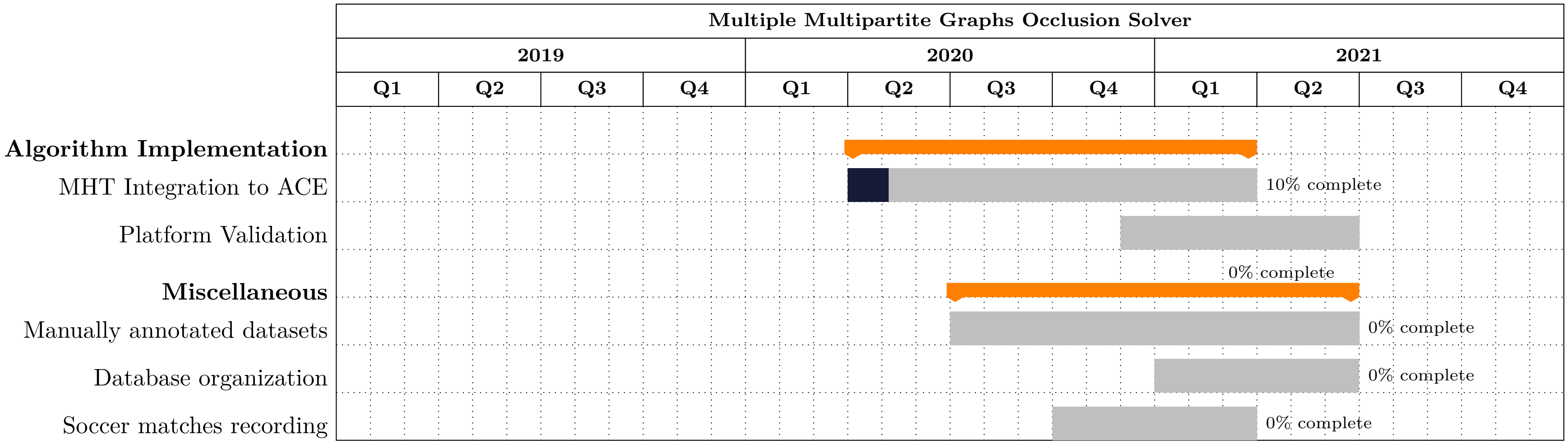
- A parallel study on players' distribution over the field is to be performed to best implement the algorithm by recognizing the frames where occlusion is dense and requires Multiple Hypothesis Tracking (MHT) for solving the assignments (occlusion detection).

3. Validation of tracking tool

- The validation will be performed using a manually annotated ground truth database generated through an Annotation Tool.
- The ground truth database currently includes the 2D position of soccer players on different videos, the tool will include a means to identify occlusion events, thus allowing to generate an occlusion ground truth database for validation.
- New ground truth database will be generated with supporting information from GPS trackers located in the uniform of the players, wore during practices.
- The main metrics to be used are the Multiple Object Tracking Precision (MOTP) and Multiple Object Tracking Accuracy (MOTA).

3.1 Time frame




Multiple Multipartite Graphs Occlusion Solver
2019
2020
2021
Q1
Q2
Q3
Q4
Q1
Q2
Q3
Q4
Q1
Q2
Q3
Q4
Algorithm Implementation
MHT Integration to ACE
10% complete
Platform Validation
0% complete
Miscellaneous
Manually annotated datasets
0% complete
Database organization
0% complete
Soccer matches recording
0% complete

Chapter 4

Preliminary Results

This chapter displays the ongoing efforts on defining the occlusionevents space and detection, which is key in developing an efficient occlusionsolver.

4.1 Detecting Occlusion

Two main approaches are considered in order to identify scenes with occlusionevents, the first one is to check for missing labels or inconsistent nodes in the constructed multipartite graph of a frames sequence. This method, considering that all the blobs are actually soccer players, allows to exactly identify the frames where the tracking algorithm is failing and thus highlights were it needs a fix, and by analyzing the blobs history throughout the sequence it is possible to also identify the cause of the failure. However, this method requires an analytical step, usually performed by a human with knowledge and understanding of the tracking algorithm and failure modes, but also requires the multipartite graph to be constructed, which implies that the blobs have already been labeled. It will require an automatic semantic interpreter of the multipartite graphs to clearly classify which failures are due to occlusionand which due to other causes, and just after that is done an occlusionsolver may make sense. It is not the intention of this research to develop a programmed module for the semantic interpretation of failure modes during a tracking process, nor to wait until the failure has occurred to fix it, but to be able to detect an scene with high probability of occlusion(high probability of the tracking algorithm to fail due to occluding players) before the players are even identified, but already detected.

Therefore, the second approach discards knowing which player is which and just requires the cleaned blob's positions. Considering this focus will allow to classify scenes according to how probable it is for an occlusionevent to cause a tracking failure, making it possible to have a selection of tracking algorithms and, based on the scene type and other sights its behaviour, pick the algorithm that works best. Having the previous implementation trait in mind, an occlusionsolver would be a module that can be enabled/disabled automatically during a tracking process to improve the results on a selected sequence of frames.

This research looks forward to the second mentioned approach, as it adds flexibility and is expected to add robustness by allowing a *smart* algorithm selection, as typical implementations marry a single tracking algorithm, despite many having proven to be more efficient on different situations than others [31].

4.2 Soccer Player's Spatial Distribution

By definition, occlusionhappens when the objects of interest are close to each other, be it partial of complete occlusion. Using this hint, it is expected that a scene with high probability of occlusionwould be one in which soccer players are too close to each other, an agglomerated image frame. Therefore, it is paramount to have a measure of this agglomeration in order to identify an image frame as with high complexity or low complexity for the algorithm tracker to solve due to occlusions.

To do so, this research pushes towards defining a quantitative metric that can relate occlusion probability from soccer player's spatial distribution over the field [40].

Scene Classification

Taking into account soccer players agglomeration, scenes can be classified as having high agglomeration, moderate agglomeration and low agglomeration, being the first one a frame in which many players are closed together, and the latter one where players are spaced, labeled as Bad-Behaved(high agglomeration), Medium-Well-Behaved(medium agglomeration) and Well-Behaved(low agglomeration) scenes (figure 4.3). Therefore, it is expected to encounter more occluded players and higher Occlusion Event Complexity (OEC) in a Bad-Behavedscene than in a Medium-Well-Behavedand a Well-Behavedscenes.

There are different ways in which an occlusion can occur, also, it is possible to have many clusters of occluded players distanced from each other, like in figure 4.1 with at least 5 clusters identified inside the soccer field, the current classification still needs more refinement in order to better reflect the possible scenarios, as well as the definition of the OEC.

Figure 4.1: Identification of occlusion clusters in a Bad-Behavedscene frame.

Players Dispersion Level measurement

According to [8], [9], [17], there exist three main metrics to study the soccer player's dispersion over the field, Stretch Index, Surface Area, and Team Length. Team Length requires knowing to which team each player belongs, as it is used to describe a team's tactics and positioning over the field. Surface Area metrics may or may not require team belonging information, it is used to describe areas where players step the more into, this to reflect concurrent field zones for different key plays and with team belonging data it can show how well a team executes a maneuver by showing where the players moved. Stretch Index compares the average distance of a team's player and geometrical center, being a measure of how compact or stretched is the team over the field. Although these metrics are commonly used over tracked or identified players and teams (after the final and manually cleaned tracking results are obtained), evaluating over their base idea can be useful while using blobs, as the main piece of information used are the objects position in the field.

To represent the players dispersion over the field, two relative distances are considered: the distance between players and the distance from a global reference point. There may be distanced clusters of occluded players over the field, the global reference can depict this case. There may be an agglomeration of many players in a close area (like in a corner kick) but the players are spaced enough to not occlude, the distance between players depicts this case. From the previous mentioned metrics, the Stretch Index is the one which expression matches best the above needs.

Taking $P_{axis,n}$ as the n'th player's position over $axis = \{X, Y\}$ in frame k, $GC_{axis}(k)$ the position of the geometrical center of the team in frame k, with N the total number of team players in frame k, Stretch index (SI) can be calculated by equation 4.1:

$$SI = \frac{\sum_n^N \sqrt{(P_{x,n}(k) - GC_x(k))^2 + (P_{y,n}(k) - GC_y(k))^2}}{N} \tag{4.1}$$

The previous expression has the limitation that it requires to have each team's players identified (and tracking cleaned) in order to obtained $GC_{axis}(k)$. However, if it is ignored that there exist three main cluster of players (two teams and umpires), but a whole big cluster of blobs to track, and the geometrical center $GC_{axis}(k)$ is seen as the center of that cluster of objects, SI is, therefore, the magnitude of the average of the distances blob-to-cluster, which describes the first distance desired for the metric to design, the distance between players. If $GC_{axis}(k)$ is replaced by a general reference point, equation 4.1 can be rewritten to take the reference's position as a parameter R:

$$SI(R) = \frac{\sum_n^N \sqrt{(P_{x,n}(k) - R_x(k))^2 + (P_{y,n}(k) - R_y(k))^2}}{N} \tag{4.2}$$

Furthermore, if the general reference R does not move throughout the match recording, like the four corners of the soccer field delimiters, it becomes a merely constant that can be obtained from the early stages of the tracking process, in the case of panoramic recordings of the whole field where the four corners are visible and with constant camera perspective.

$$SI(R) = \frac{\sum_n^N \sqrt{(P_{x,n}(k) - R_x)^2 + (P_{y,n}(k) - R_y)^2}}{N} \tag{4.3}$$

To comply with the desired convention, blobs are to be referenced as *players* even though a blob has not yet been identified as one, thus, distances are referred as player-to-cluster, player-to-reference, player-to-player instead of blob-to-cluster- blob-to-reference and blob-to-blob.

Figure 4.2 depicts what $SI(R)$ represents, where the red lines are the vectors player-to-reference, and the blue line is the average vector, where $SI(R)$ is the blue line's magnitude. This measure represents the position in the field where the most players are agglomerated.

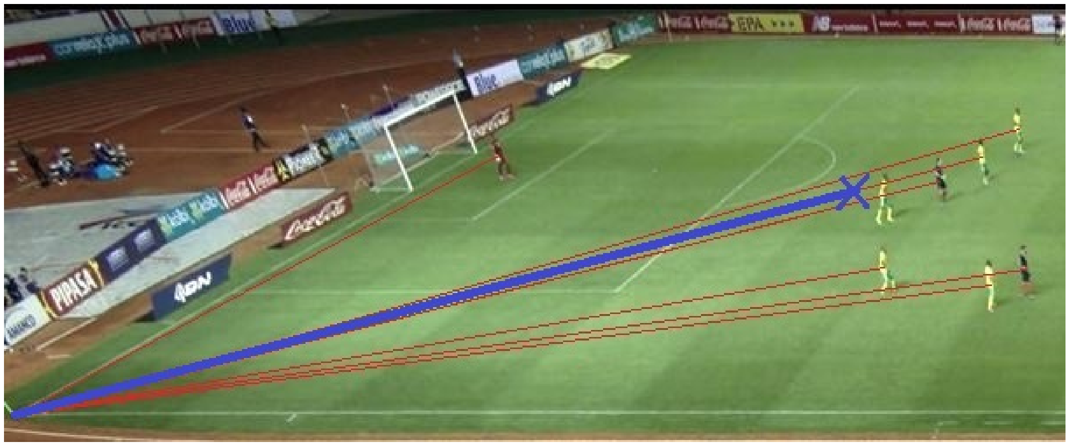


Figure 4.2: Graphical description of the modified Stretch Index with a general reference SI(R), using the corner of the field as the reference. Red lines are the vectors player-to-reference and blue line is the average vector. SI(R) is the magnitude of the average vector.

Having identified the average vector, i.e. the place with more agglomeration, the next step is to convert its magnitude into a metric of how agglomerated are the players around that point. To accomplish this, the standard deviation is considered, using the equation 4.4, where $PR_n(k) = P_n(k) - R$ is the vector player-to-reference, with $P_n(k)$ the player's position taken from the origin of the image's frame (pixels 0,0), R the general reference from the same origin, $|PR_n(k)|$ the magnitude of the vector player-to-reference, and μ the mean of the vector's magnitudes.

$$\sigma = \sqrt{\frac{1}{N}\sum_{n}^{N}(|PR_n(k)| - \mu)^2} \tag{4.4}$$

Where $SI(R)$ can replace μ, thus the standard deviation σ of the player-to-reference vectors' magnitude around the average vector magnitude is described by equation 4.5.

$$\sigma = \sqrt{\frac{1}{N}\sum_{n}^{N}(|PR_n(k)| - SI(R))^2} \tag{4.5}$$

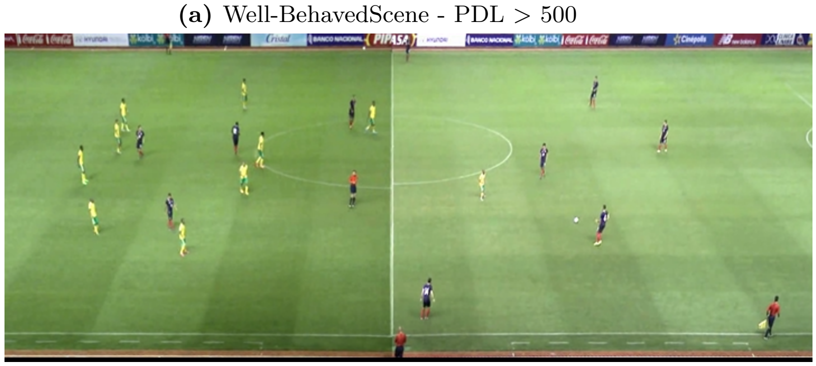
If σ is low, then the players are gathered close around the agglomeration point, if it is high, the players are distant from the agglomeration point. In this sense, a low σ would mean a high occlusion probability due to the players being close to the same point.

Players position affect the shape of the cluster, depending on the reference taken into account, the average vector is dependent on its reference, to include this information it is considered an average of the σ using each corner of the field, finally defining the Players Dispersion Level (PDL) metric, described by equation 4.6, where $c = 1, 2, 3, 4$ is the corner used as reference and R_c is the corner's position.

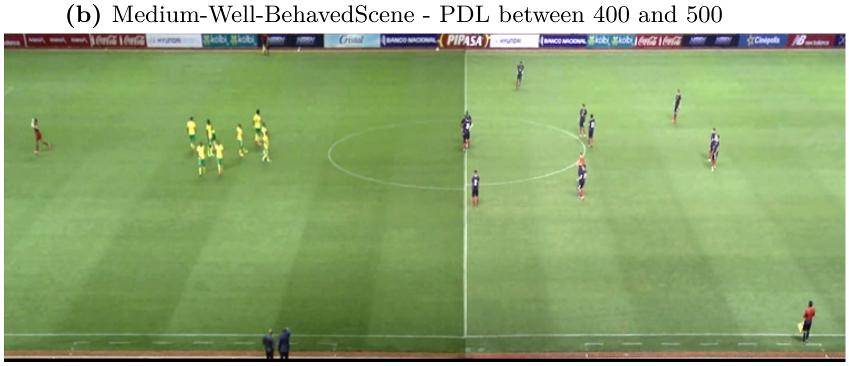
$$PDL = \frac{\sum_c^4 \sqrt{\frac{1}{N}\sum_n^N (|PR_n(k)| - SI(R_c))^2}}{4} \tag{4.6}$$

Testing the metric with a short video sequence of 40 s, three main value regions can be extracted that match with the required scenes classification, were a Bad-Behavedhas a low PDL and Well-Behavedhas a high PDL, as seen in figure 4.3, for scene classification, and figure 4.4 for the standard deviations with each corner as reference and the final PDL calculation for each video frame.

(a) Well-BehavedScene - PDL $>$ 500

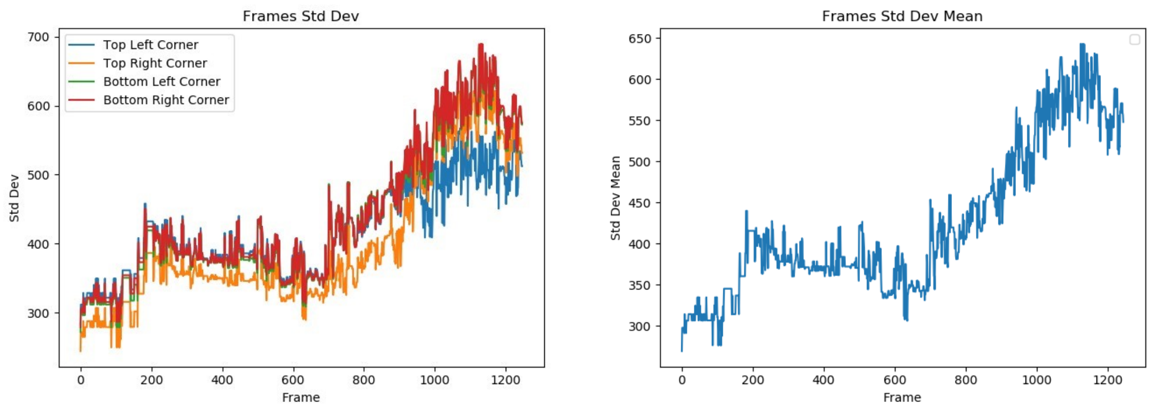


(b) Medium-Well-BehavedScene - PDL between 400 and 500



(c) Bad-BehavedScene - PDL $<$ 400

Figure 4.3: Soccer player match scenes classification by player's agglomeration



(a) Standard deviation for each corner reference

(b) Average of standard deviations

Figure 4.4: Players Dispersion Level of a 40 s clip from a soccer match using the four corners of the soccer field as references

Accurate thresholds to define each class are still pending, as it requires longer recordings and different matches. The bigger limitation noticed so far is that this metric takes all the blobs as a whole single cluster represented by the average vector, where multiple smaller clusters of occluded players may remain hidden if well spaced from each cluster. Also, as the metric is designed to use blob's positions disregarding player identification stage, there may exist spurious blobs that affect the metrics by adding false players' positions into account to the calculation. Also, this metric is a single value that encompasses the whole scene rather than an actual distribution of the players' agglomeration over the field, keeping its core intention from the Stretch Index.

Kernel Density Estimation for Spatial Distribution Function

Another approach on describing soccer players' spatial distribution is to define an actual distribution function with which to identify field regions with the highest probability of players being agglomerated, thus being a direct metric of occlusion probability.

Soccer players motion complexity has proven to require tracking algorithms designed for non-Gaussian non-linear problems [12], [39] in order to improve the results, as predicting where a player may move is difficult even knowing the team's strategic plan, although it is more probable that the player will continue forward following its inertia, humans can quickly change direction and velocity, as well as drifting on the grass. Thus, a well fitted motion model is not yet developed and rather represented as a complex net of possible actions a player can take in different situations, specially for soccer simulations where virtual assets are expected to behave as a real player, studying teams strategy instead [1], [48]. Some other studies try to fit goals, team strategy, passes, attack/defense and other in-game actions to different probabilistic functions [34], [56], however, these studies are focused on analysing game strategy and statistics, how players interact and react, and what differentiate elite teams and players from the rest, using clean tracking results. Main metrics to describe players distribution over the field remains Stretch Index and Surface Area [8].

Kernel Density Estimation is a non-parametric statistical tool that allows to estimate the probability density function of a random variable, therefore, having the series of n observations $x_1, x_2, ..., x_n$ taken from the population X with an unknown probability distribution function $f(x)$, the kernel estimate $\hat{f}(t)$ of original $f(x)$, with a kernel function $K(x_i, t)$, is defined by the equation 4.7, following convention and implementation details as shown in [65].

$$\hat{f}(t) = \frac{1}{n}\sum_{i=1}^{n} K(x_i, t) \tag{4.7}$$

Considering the current scenario with independent populations X, Y the players positions in the frame k, with pair observations $x = x_1, x_2, ..., x_n$ and $y = y_1, y_2, ..., y_n$, bivariate kernel estimation $\hat{f}(x, y)$ can be described by equation 4.8, where kernel function $K(x_i, t) = \frac{1}{h}K(\frac{x-t}{h})$, due to its symmetry property [65], with h the smoothing parameter.

$$\hat{f}(x, y) = \frac{1}{nh_x h_y}\sum_{i=1}^{n} K(\frac{x_i - x}{h_x}, \frac{y_i - y}{h_y}) \tag{4.8}$$

Using blobs positions from a 40 s clip of a soccer match and the Python library Seaborn [64], with a Gaussian kernel function and smoothing parameter calculated by the Seaborn's function *kdeplot* internal calculations, it is possible to generate a 2D plot of the bivariate estimated probabilistic function for the players' positions distribution over the field for each frame of the clip (see figure 4.5).

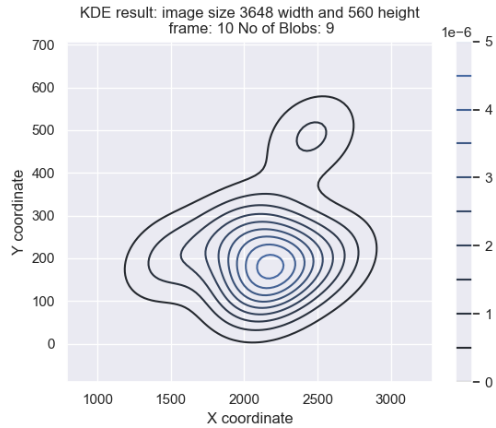
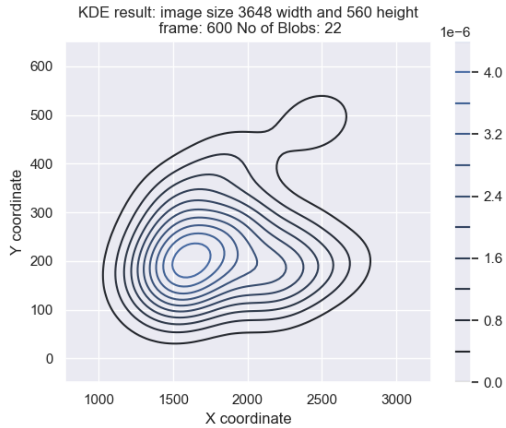


Figure 4.5: Estimated bivariate probabilistic function of soccer players spatial distribution for frame $k = 10$

Comparing results with PDL, it can be seen that they coincide in which scenes players are more likely agglomerated and in which they are spaced. Fro frame $k = 600$ the $PDL < 400$, which means it's basically a Bad-Behavedscene, which can be seen in the KDE plot by a stretched shape (figure 4.6), while the frame $k = 1100$ with a $PDL > 500$ has a more elongated shape with lower peaks, being a Well-Behavedscene (figure 4.7).

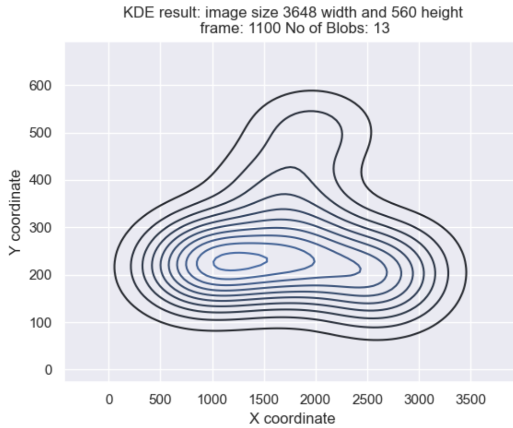
This method presents an advantage over the PDL one as it allows to identify probabilistic regions instead of a global value, thus it highlights clustered areas, being useful to classify scenes for its general agglomeration density and also to distinguish image regions of high agglomeration, allowing to detect independent clusters like the one located at (x=2500, y=500) in figures 4.6a and 4.7a, providing a tool to not only select frames to use the occlusion solver on, but also to select image regions for a more localized and thorough occlusion solver, if desired.



(a) Estimated bivariate probabilistic function

(b) Respective frame from soccer match clip

Figure 4.6: Estimated bivariate probabilistic function of soccer players spatial distribution for frame $k = 600$ and $PDL < 400$, classified as Bad-Behavedscene



(a) Estimated bivariate probabilistic function

(b) Respective frame from soccer match clip

Figure 4.7: Estimated bivariate probabilistic function of soccer players spatial distribution for frame $k = 1100$ and $PDL > 500$, classified as Well-Behavedscene

Next steps require to analyze the different players' dispersion shapes generated by KDE, to actually classify scenes studying the generated probabilistic function.

Bibliography

[1] E. Alfaro, “Optimización del componente de inteligencia artificail del simulador simple soccer”, Bachelors Degree graduation project, Master’s thesis, Universidad de Costa Rica, Sep. 2049.

[2] A. Amditis, G. Thomaidis, P. Maroudis, P. Lytrivis, and G. Karaseitanidis, “Multiple hypothesis tracking implementation”, in, ser. Laser Scanner Technology. IntechOpen, 2012, ch. 10.

[3] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications.* ArtechHouse, 096483122, Nonvood, MA, 1990.

[4] L. Bazzani, D. Bloisi, and V. Murino, *A comparison of multi hypothesis kalman filter and particle filter for multi-target tracking*, 2009.

[5] T. Bebie and H. Bieri, “Soccerman-reconstructing soccer games from video sequences”, in *1998 International Conference on Image Processing. ICIP98*, 1998. DOI: 10.1109/ICIP.1998.723665.

[6] S. Blackman, “Multiple hypothesis tracking for multiple target tracking”, in *IEEE Aerospace and Electronic Systems Magazine*, vol. 19(1), 2004s, pp. 5–18. DOI: 10.1109/maes.2004.1263228.

[7] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Online multi-person tracking-by-detection from a single uncalibrated camera”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, 2011, pp. 1820–1833. DOI: 10.1109/TPAMI.2010.232.

[8] F. M. Clemente, M. S. Couceiro, F. M. L. Martins, and R. Mendes, “An online tactical metrics applied to football game”, *Research Journal of Applied Sciences, Engineering and Technology*, vol. 5(5), pp. 1700–1719, 2013. DOI: 10.19026/rjaset.5.4926.

[9] F. Clemente, J. Sequeiros, A. Correira, F. Sylva, and F. Laurenço, “Measuring the dispersion of the players”, in, ser. SpringerBriefs in Applied Sciences and Technologies. Springer, 2018, pp. 43–53. DOI: doi.org/10.1007/978-3-319-59029-5_5.

[10] I. J. Cox and S. Hingorani, “An efficient implementation of reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18(2), 1996, pp. 138–150. DOI: 10.1109/34.481539.

[11] J. Czyz, B. Risti, and B. Macq, “A color-based particle filter for joint detection and tracking of multiple objects”, in *IEEE International Conference in Acoustics, Speech, and Signal Processing*, vol. 2, 2005.

[12] A. Dearden, Y. Demiris, and O. Grau, “Tracking football player movement form a single moving camera using particle filters”, in *CVMP-2006*, 2006, pp. 29–37.

[13] A. Doucet and A. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later”, in *The Oxford Handbook of non-linear filtering*. OUP Oxford, 2012, pp. 656–704. DOI: 10.1.1.157.772.

[14] FIFA, *Dispositivos de seguimiento electrónico del rendimiento*, Last accessed July 2019. [Online]. Available: https://football-technology.fifa.com/es/media-tiles/epts1/.

[15] P. Figueroa, N. Leite, and R. M. L. Barros, “Tracking soccer players aiming their kinematical motion analysis”, *Computer Vision and Image Understanding*, vol. 101, pp. 122–135, 2006. DOI: 10.1016/j.cviu.2005.07.006.

[16] P. Figueroa, N. Leite, R. M. L. Barros, I. Cohen, and G. Medioni, “Tracking soccer players using the graph representation”, in *17th International Conference on Pattern Recognition*, vol. 4, 2004.

[17] H. Folgado, K. Lemmink, W. Frencken, and J. Sampaio, “Length, width and centroid distance as measures of teams tactical performance in youth football”, *European Journal of Sport Science*, vol. 14(1), pp. 487–492, 2012. DOI: 10.1080/17461391.2012.730060.

[18] E. Fortunato, W. Kreamer, S. Mori, C. Chee-Yee, and G. Castanon, “Generalized murty’s algorithm with application to multiple hypothesis tracking”, in *Proceedings of 2007 10th International Conference on Information Fusion*, 2007. DOI: 10.1109/icif.2007.4408017.

[19] S. Gedikli, J. Bandouch, N. Hoyningen-Huene, B. Kirchlechner, and M. Beetz, “An adaptive vision system for tracking soccer players from variable camera settings”, in *Fifth International Conference on Computer Vision Systems*, 2007.

[20] R. Hess and A. Fern, “Discriminatively trained particle filters for complex multi-object tracking”, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 240–247. DOI: 10.1109/CVPR.2009.5206801.

[21] Z. L. Inc., *Zepp play - soccer*, Last accessed Jan 2020. [Online]. Available: http://www.zepp.com/en-us/soccer/.

[22] H. Itoh, T. Takiguchi, and Y. Ariki, “3d tracking of soccer players using time-situation graph in monocular image sequence”, in *21st International Conference on Pattern Recognition*, 2012.

[23] S. Iwase and H. Saito, “Parallel tracking of all soccer players by integrating detected positions in multiple view images”, Department of Information and Computer Science, Keio University, Japan, Tech. Rep., 2004.

[24] S. Jiang H. Fels and J. Little, “A linear programming approach for multiple object tracking”, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. DOI: `10.1109/cvpr.2007.383180`.

[25] S.-W. Joo and R. Chellapa, “A multiple-hypothesis approach for multiobject visual tracking”, in *IEEE Trans. Image Process*, vol. 16(11), 2007, pp. 2849–2854. DOI: `10.1109/tip.2007.906254`.

[26] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, “Multiple hypothesis tracking revisited”, in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. DOI: `10.1109/iccv.2015.533`.

[27] Kizanaro, *Kizanaro*, Last accessed May 2019. [Online]. Available: `https://cie.ort.edu.uy/35730/18/kizanaro.html`.

[28] A. Koutsia, N. Grammalidis, K. Dimitropoulos, M. Karaman, and L. Goldmann, “Football player tracking from multiple views - using a novel background segmentation algorithm and multiple hypothesis tracking”, in *Proceedings of the Second International Conference on Computer Vision Theory and Applications*, INSTICC, SciTePress, 2007, pp. 523–526. DOI: `10.5220/0002051205230526`.

[29] M. Kristan, J. Pers, M. Perse, and S. Kovacic, “Closed-world tracking of multiple interacting targets for indoor-sports applications”, in *Computer Vision and Image Understanding*, vol. 113, 2009, pp. 598–611. DOI: `10.1016/j.cviu.2008.01.009`.

[30] S. Lefèvre, C. Fluck, B. Maillard, and N. Vincent, “A fast snake-based method to track football players”, in *IAPR Workshop on Machine Vision Applications*, The University of Tokyo, Japan, 2000.

[31] M. Manafifard, H. Ebadi, and H. A. Moghaddam, “A survey on player tracking in soccer videos”, *Computer Vision and Image Understanding*, vol. 159, pp. 19–46, 2017. DOI: `10.1016/j.cviu.2017.02.002`.

[32] ——, “Appearance-based multiple hypothesis tracking: Application to soccer broadcast videos analysis”, *Signal Processing: Image Communication*, vol. 55, pp. 157–170, 2017. DOI: 10.1016/j.image.2017.04.001.

[33] MatchAnalysis, *Match analysis – unrivaled video and statistical analysis for soccer (football)*, Last accessed May 2019. [Online]. Available: http://matchanalysis.com/index.htm.

[34] I. McHale and P. Scarf, “Modelling soccer matches using bivariate discrete distributions with general dependence structure”, *Statistica neerlandica*, vol. 61(4), pp. 432–445, 2007. DOI: doi.org/10.1111/j.1467-9574.2007.00368.x.

[35] T. Misu, M. Naemura, W. Zheng, Y. Izumi, and K. Fukui, “Robust tracking of soccer players based on data fusion”, in *16th International Conference on Patter Recognition*, vol. 1, 2002.

[36] A. Mora, “Cell phenotype classification using m-phase features in live-cell bright field time-laps microscopy”, Master’s thesis, Universidad de Costa Rica, 2018.

[37] E. Morais, A. Ferreira, S. A. Cunha, R. M. Barros, A. Rocha, and S. Goldenstein, “A multiple camera methodology for automatic localization and tracking of futsal players”, *Pattern Recognition Letters*, vol. 39, pp. 21–30, 2014.

[38] G. N.J., S. D.J., and S. A.F.M., “Novel approach to nonlinear/non-gaussian bayesian state estimation”, in *F – Radar and signal Processing*, vol. 140, 1993, pp. 107–113. DOI: 10.1049/ip-f-2.1993.0015.

[39] K. Nummiaro, E. Koller-Meier, and L. Van Gool, “An adaptive color-based particle filter”, *Image and Vision Computing*, vol. 21(1), pp. 99–110, 2003. DOI: doi.org/10.1016/S0262-8856(02)00129-4.

[40] L. Núñez-Meoño, M. Villalta, and F. Siles-Canales, “Initial approachon soccer match’sscene classification by players’field spatial distribution”, *Tecnología en Marcha*, vol. 33(5), pp. 60–65, 2020. DOI: doi.org/10.18845/tm.v33i5.5077.

[41] H.-W. Ok, Y. Seo, and K.-S. Hong, “Multiple soccer players tracking by condensation with occlusion alarm probability”, IIP Lab., Pohang University of Science, and Technology (POSTECH), Republic of Korea, Tech. Rep., 2002.

[42] *Paint, sports analysis tool*, Last accessed Aug 2019, ChyronHego. [Online]. Available: https://www.thebroadcastbridge.com/content/entry/11407/chyronhego-paint-7.4-provides-powerful-sports-telestration-and-analysis.

[43] R. Radakovic, M. Dopsaj, and R. Vulovic, “The reliability of motion analysis of elite soccer players during match measured by the tracking motion software system”, in *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2015. DOI: `10.1109/BIBE.2015.7367676`.

[44] J. Rollin, R. Giulianotti, P. Christopher, R. Weil, and B. Joy, *Football soccer*, Last accessed May 2019. [Online]. Available: `https://www.britannica.com/sports/football-soccer/Play-of-the-game`.

[45] K. H. Rosen, *Discrete Mathematics and Its Applications*, 5th ed. McGraw-Hill, 2003.

[46] H. Sabirin, H. Sankoh, and S. Naito, “Automatic soccer player tracking in single camera with robust occlusion handling”, in *IEICE Transaccions on Information and Systems*, vol. E98-D, 2015, pp. 1580–1588. DOI: `10.1587/transinf.2014edp7313`.

[47] B. Sahbani and W. Adiprawita, “Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system”, in *2016 6th International Conference on System Engineering and Technology*, 2016.

[48] R. Sałustowicz, M. Wiering, and J. Schmidhuber, “Learning team strategies: Soccer case studies”, *Machine Learning*, vol. 33, pp. 263–282, 1998. DOI: `doi.org/10.1023/A:1007570708568`.

[49] M. Sanjeev, B. Ristic, N. Gordon, and T. Mansell, “Bearings-only tracking of manoeuvering targets using particle filters”, *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2351–2365, 2004.

[50] Y. Seo, S. Choi, H. Kim, and K.-S. & Hong, “Where are the ball and players? soccer game analysis with color-based tracking and image mosaick”, in *9th International Conference on Image Analysis and Processing*, vol. II, Springer-Verlag, 1997, pp. 196–203.

[51] H. B. Shitrit, J. Berclaz, and F. Fleuret, “Multi-commodity network flow for tracking multiple people”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, 2014, pp. 196–203. DOI: `10.1109/tpami.2013.210`.

[52] F. Siles, “Automated semantic annotation of football games from tv broadcast”, Doctoral dissertation, Technische Universitad Munchen, 2014.

[53] F. Siles and J. C. Saborío, “Parallel spatial segmentation for the automated analysis of football”, in *5th IEEE International Workshop and Conference on Bioinspired Intelligence*, 2015.

[54] R. Singer, R. Sea, and K. Housewright, "Derivation and evaluation of improved tracking filter for use in dense multitarget environments", in *IEEE Trans. Information Theory*, vol. 20(4), 1974, pp. 423–432. DOI: 10.1109/tit.1974.1055256.

[55] S. Soomro K. Khokhar and M. Shah, "Tracking when the camera looks away", in *2015 IEEE International Conference on Computer Vision Workshop*, 2015.

[56] W. Spearman, A. Basye, G. Dick, R. Hotovy, and P. Pop, "Physics-based modeling of pass probabilities in soccer", in *MIT Sloan Sports Analytics Conference*, 2017.

[57] M.-L. Sport, *Mc-lloyd products*, Last accessed July 2019. [Online]. Available: http://mac-lloyd.com/en/products/.

[58] STATS, *Stats player tracking*, Last accessed May 2019. [Online]. Available: https://www.stats.com/player-tracking/.

[59] ——, *Stats sportvu football player tracking*, Last accessed May 2019. [Online]. Available: https://www.stats.com/sportvu-football/.

[60] A. algorithm for tracking multiple targets, "Football player tracking from multiple views - using a novel background segmentation algorithm and multiple hypothesis tracking", in *Proceedings of IEEE Transactions on Automatic Control*, vol. E98-D, 1979, pp. 423–432. DOI: 10.1109/tac.1979.1102177 .

[61] *Tracab optical tracking, product information sheet*, Last accessed May 2019, ChyronHego. [Online]. Available: https://chyronhego.com/wp-content/uploads/2019/01/TRACAB-PI-sheet.pdf.

[62] M. Villalta, "Football players tracking using multipartite graphs with ultra high definition video", Master's thesis, Universidad de Costa Rica, 2019.

[63] M. Villalta and F. Siles, "Parallelization of a multipartite graph matching algorithm for tracking multiple football players", in *Fifth International Conference on Parallel, Distributed and Grid Computing*, 2018. DOI: 10.1109/pdgc.2018.8745720.

[64] M. Waskom, *An introduction to seaborn*, Last accessed June 2020. [Online]. Available: https://seaborn.pydata.org/introduction.html.

[65] S. Węglarczyk, "Kernel density estimation and its application", in *ITM Web of Conferences, XLVIII Seminar of Applied Mathematics*, vol. 23(00037), 2013, pp. 1700–1719. DOI: 10.1051/itmconf/20182300037.

[66] L. Wei-Lwun, T. Jo-Anne, J. Little, and K. Murphy, "Learning to track and identify players from broadcast sports videos", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, 2019.

[67] R. Wood, *World's most popular sports by fans*, Last accessed May 2019, Topend Sports. [Online]. Available: `https://www.topendsports.com/world/lists/popular-sport/fans.htm`.

[68] M. Xu, J. Orwell, and G. Jones, "Tracking football players with multiple cameras", Digital Imaging Research Centre, Kingston University, Kingston upon Thames, KT1 2EE, UK, Tech. Rep., 2004.

[69] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey", *ACM Computing Surveys*, vol. 38, no. 13, pp. 1–16, 2004. DOI: `10.1145/1177352.1177355`.

[70] K. Yoon, Y. Song, and M. Jeon, "Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views", in *IET Image Processing*, vol. 12(7), 2018, pp. 1175–1184.

[71] D. A. Zaugg, A. A. Samuel, D. E. Waagen, and H. A. Schmitt, "A comparison of particle filters and multiple-hypothesis extended kalman filters for bearings-only tracking", in *Acquisition, Tracking, and Pointing XVIII*, vol. 5430, 2004. DOI: `10.1117/12.541625`.

Appendices

Appendix A

Classification of tracking methods

The following table illustrates the tracking methods according to the type of method of the core algorithm (Deterministic, Probabilistic, Data Association) and if it takes into consideration occlusion handling or not. This distribution is based on the review paper by Yilmaz et all. [69], its generalization by Villalta, M. [62] and current review of professional literature.

	Deterministic Methods	Probabilistic Methods	Data Association Methods
No Occlusion Handling nor Detection	**Template Matching**	**Particle Filter**	**Graph Representation**
	Seo, Choi, Kim, *et al.* [50]	Czyz, Risti, and Macq [11]	Wei-Lwun, Jo-Anne, Little, *et al.* [66]
	Bebie and Bieri [5]	Dearden, Demiris, and Grau [12]	Soomro and Shah [55]
	Lefèvre, Fluck, Maillard, *et al.* [30]		Siles [52]
Occlusion Handling/Detection	**Feature Mixture**	**Particle Filter ++**	**Multiview**
	Misu, Naemura, Zheng, *et al.* [35]	Kristan, Pers, Perse, *et al.* [29]	Xu, Orwell, and Jones [68]
	Sabirin, Sankoh, and Naito [46]	Hess and Fern [20]	Figueroa, Leite, Barros, *et al.* [16]
		Itoh, Takiguchi, and Ariki [22]	**Graph Representation**
		Morais, Ferreira, Cunha, *et al.* [37]	Figueroa, Leite, and Barros [15]
		Occlusion Alarm Probability	Villalta [62]
		Ok, Seo, and Hong [41]	**Multi-layer Graph**
			Jiang and Little [24]
			Shitrit, Berclaz, and Fleuret [51]
			Multiple Hypothesis
			Gedikli, Bandouch, Hoyningen-Huene, *et al.* [19]
			Breitenstein, Reichlin, Leibe, *et al.* [7]
			Koutsia, Grammalidis, Dimitropoulos, *et al.* [28]
			Manafifard, Ebadi, and Moghaddam [32]

Table A.1: Distribution of main references according to core tracking method (Deterministic, Probabilistic, Data Association), and grouped as *Occlusion Handling/Detection* or *No Occlusion Handling nor Detection* if occlusion events are considered/managed or not

Appendix B

Distribution of professional literature

Following plots depict the distribution over the years of the reviewed academic and professional literature (figure B.1), as well as the cumulative of references directly related to the use of Multiple Hypothesis Tracking for multiple object tracking, focused on soccer players' tracking (B.2).

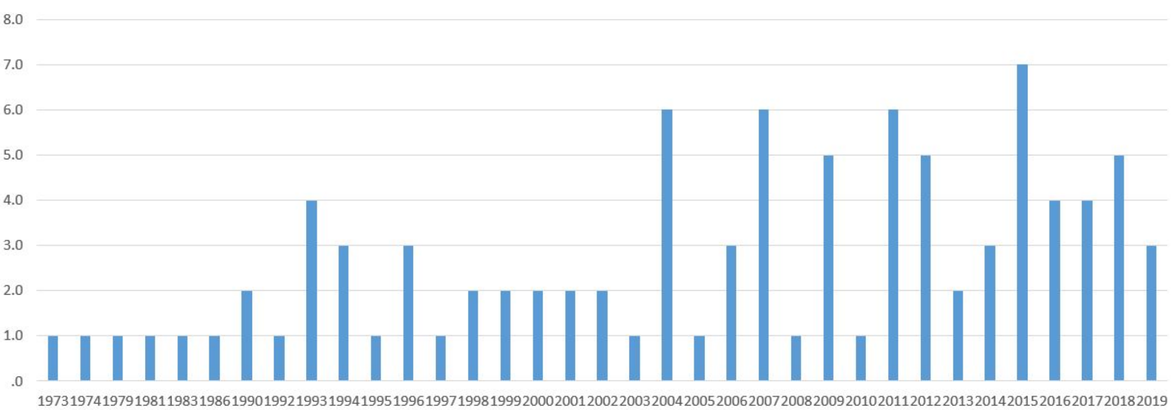


Figure B.1: Histogram of yearly distribution of professional literature related to Multiple Object Tracking

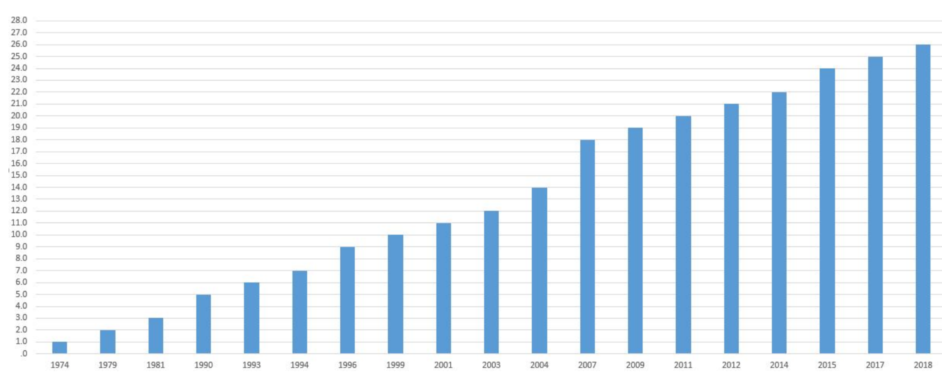


Figure B.2: Cumulative of professional literature directly related to Multiple Hypothesis Tracking