Socially-Aware Robot Navigation by Leveraging Group Detection

Schmuck, Viktor

Awarding institution:
King's College London

Socially-Aware Robot Navigation by Leveraging Group Detection

Viktor Schmuck

Supervisors: Dr. Oya Celiktutan
Dr. Matthew Howard

The Department of Engineering
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy in Robotics

July 2023

I was taught that the way of progress was neither swift nor easy.

***

Be less curious about people and more curious about ideas.

Marie Curie

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 100,000 words including appendices, the bibliography, footnotes, tables and equations.

Viktor Schmuck
July 2023

Acknowledgements

In all fairness, I did not think that I would end up working with social robotics. I was just interested in Human-Computer Interaction. The work behind this thesis began around the middle of my Master's degree. I had a bike accident commuting from a lecture to my student job and the next day my GP sent me to the hospital. I was instructed to get an X-ray, but this task proved to take the better of me in the complex building. 40 minutes, two helpful nurses and a doctor later, I was no longer lost, and I left with splints and the spark of an idea about reforming how people can be aided in navigating complex indoor environments. In the following year, I tailored my Master's Thesis to address this topic. After I graduated, my supervisors secured me a position as a Research Assistant to continue my investigation, starting my research in the area of socially-aware navigation. Wanting to earn a doctorate but not willing to give up on the research area, I pitched the idea of improving robot navigation and investigating social awareness in crowded spaces to my current supervisor. And the rest is history, presented in this thesis.

I would first like to share my sincerest appreciation for the opportunity and the support provided throughout this journey by my supervisor, Dr. Oya Celiktutan. From her, I always received the help and (much-required) criticism I needed, and her knowledge and guidance moulded me and this work into something truly better.

A special thank you to my former supervisors, Prof. Matthias Rehm and Prof. David Meredith, who guided me onto this path during and after my Master's Degree, and who played a significant role in shaping my desire to become a researcher.

In classical fashion, I would like to thank my parents and family for their support and acceptance throughout the past few years, and my friends and colleagues who believed in me and offered advice when I needed it.

Finally, I would like to say thanks to those closest to me. Lili, Dave, Tamás, Norbert. You have been there with me every step of the way. I am grateful for your support and honoured by the trust you have in me. I would like to make a special mention to Lili. I could not ask for a better person to help me get through a doctorate, let alone through a degree that has a global pandemic in the middle of it. Thank you for being 'the little things' in my days, inspiring me to progress, and for listening to my ramblings. The best part of this was sharing it with you.

Abstract

In the past years, the deployment of embodied agents has become more and more commonplace in human-populated environments. This trend poses a need for safe robot navigation with higher levels of autonomy, which is one of the key challenges within the area of embodied agent research. Previous research established reliable navigation strategies, however, they do not take into account social aspects sufficiently, which might hamper user acceptance and trust. Studies show that social awareness is an important capability that allows humans to act in an easy-to-understand and predictable manner, which are both crucial competencies in navigating crowded environments. To equip a robot with the same skills, it first needs to recognise social relations in a social setting. A part of this recognition involves the detection of interaction groups which can aid the robot in traversing crowded spaces in a socially-aware manner.

This thesis presents my research on socially-aware robot navigation in crowded indoor spaces. It has two main parts focusing on group detection and socially-aware navigation, respectively. The first part addresses the problem of group detection in crowds based on both third-person and robocentric perspectives. Throughout the thesis, there is a special emphasis on the development of robocentric solutions as this problem has not been thoroughly explored before. One of the main gaps in the literature was the absence of a publicly available benchmark dataset, which was collected from a robot's perspective, featuring dense social environments and containing annotations for groups. To address this gap, I proposed the Robocentric Indoor Crowd Analysis (RICA) dataset, which can be used for the training and evaluation of methods in the field of socially-aware robot navigation. Using the RICA dataset, I proposed two approaches to the problem of group detection. First, I devised a solution based on Agglomerative Hierarchical Clustering (AHC). This method relies on position and orientation data of detected individuals in a scene and has achieved state-of-the-art performance among unsupervised group detection methods when evaluated on common benchmark datasets as well as RICA. Second, I presented a Graph Neural Network based group detection approach, which, relying on the same input features as my AHC-based method, achieves the best performance yet among supervised solutions. Following the exploration of supervised and unsupervised group detection techniques, in the second part of this thesis, I examine the viability of using group information for

improved socially-aware navigation by integrating it into a reinforcement learning-based method, i.e., an advantage actor-critic (A2C) algorithm. In previous works, there was no agreement with regard to how social awareness of navigation methods should be evaluated. To address this gap, I presented a more thorough and unified method for benchmarking the social awareness aspect of existing solutions. This evaluation method implements the most commonly used quantitative metrics and includes a qualitative assessment as well as a Navigation Turing Test.

Taken together, the techniques and findings documented in this thesis present state-of-the-art group detection methods that are applicable to both third-person and robocentric data, and showcase why and how group information can be utilised for socially-aware navigation and the importance it bears.

Table of contents

List of figures

List of tables

Glossary

bounding box A rectangle surrounding an object to specify its position in an image. 10–13, 16, 33, 34, 38, 44, 56–58, 60, 65–72, 78–80, 85, 90, 91, 115, 117

centroid The geometric centre of an object. In the case of groups, the centre point of an o-space. 14, 45, 46, 65, 67, 78, 79, 97, 98, 103, 112, 115, 116, 118

F_1-score Also known as F-measure, F-score. The F_1-score is the harmonic mean of measured precision and recall values, where precision is the number of true positives divided by the number of both true and false positives, and recall is the number of true positives divided by true positives and false negatives. 14, 16, 41, 42, 49, 74, 85, 86, 88–92, 115

RGBD Red, Green, Blue, and Depth image data collected by an RGBD camera. Such an image carries per-pixel Depth information for each RGB pixel in an image [125]. 11, 24, 33, 49, 56, 59, 79

robocentric Data captured from a first-person, egocentric view of a robot. 10, 13, 16, 24, 26, 27, 31, 33–36, 41, 43, 48, 49, 51, 55, 59, 60, 64, 66, 71–75, 78–80, 84, 85, 92, 94, 114–117

SALSA-CPP The Cocktail Party subset of the Synergetic sociAL Scene Analysis dataset [5]. 10, 14–17, 32, 53, 84, 87–89, 91, 100, 102, 104–112

SALSA-PS The Poster Session subset of the Synergetic sociAL Scene Analysis dataset [5]. 10, 14–17, 32, 33, 74, 83, 84, 86, 88, 89, 91, 100, 102, 104–112

Synth Synthetic data collected by Cristani et al. [31] comprised of 100 situations provided by a psychologist where in each scene some individuals are free-standing, and some are forming groups. 41, 42

Acronyms

A2C Advantage Actor-Critic. 101, 102, 104, 105, 113, 116

A3C Asynchronous Advantage Actor-Critic. 45, 96, 101, 110

AHC Agglomerative Hierarchical Clustering. 26, 27, 38, 50, 64–66, 69, 78, 90, 93, 115, 116

AMI Adjusted Mutual Information. 42

AUC area under the curve. 42

CB CoffeeBreak. 42

CH score Calinski and Harabasz Score. 69

CNN convolutional neural network. 44

CP CocktailParty. 42

DB criterion Davies-Bouldin criterion. 65, 69

DoF degrees of freedom. 52

DS Dominant Sets. 38–40, 76

GCFF Graph Cuts for F-formations. 16, 38, 74, 78, 84, 89, 93

GNN Graph Neural Network. 26, 27, 39, 46, 50, 73, 75–78, 80, 84, 87, 93, 115, 118

GROWL GROup detection With Link prediction. 13, 14, 16, 78, 81, 83–85, 87–94, 115

GT ground truth. 11, 12, 14, 42, 59, 61, 66, 67, 69–71, 84, 85, 90, 92, 93, 104, 119

GTM Global Tolerant Matching. 42

HOG Histogram of Oriented Gradients. 12, 66, 67

HRI Human-Robot Interaction. 22, 98

HSR-B Toyota Human Support Robot series B. 10, 11, 52–54, 59

HVFF Hough Voting for F-formations. 38

iGROWL Improved GROup detection With Link prediction. 16, 87, 89, 91, 93, 94, 115, 117

IMU Inertial Measurement Unit. 52–54, 59, 60

IoU Intersection-over-Union. 12, 33, 66, 67, 70, 85

IPD IDIAP PosterData. 42

IRL Inverse Reinforcement Learning. 45, 47, 49

IRPM Inter-Relation Pattern Matrix. 41

JRDB JackRabbot Dataset and Benchmark. 10, 34–36, 54, 59, 60, 71

LiDAR Light Detection and Ranging. 24, 34, 44, 45, 52–54, 59, 60, 80

MAE Mean Average Error. 12, 42, 49, 65, 69–71

MLP Multi-Layer Perceptron. 13, 82–84, 93

NLVO non-linear velocity obstacle. 43

NMI Normalized Mutual Information. 42

NMS non-maxima suppression. 66

NTT Navigation Turing Test. 15, 26, 28, 48, 49, 108, 110, 113, 116

PSI Perceived Social Intelligence. 17, 48, 107–113, 118–120

REFORM REcognize F-FORmations with Machine learning. 16, 39, 73–75, 78, 84, 89, 93

ReLU Rectified Linear Unit. 13, 82, 83

RICA Robocentric Indoor Crowd Analysis. 10–14, 16, 24, 51–56, 59, 60, 62, 63, 66, 67, 69, 71, 72, 78, 79, 84, 85, 87–92, 114, 115

RL Reinforcement Learning. 26, 30, 45–47, 49, 95, 96, 99, 101, 102, 110, 118, 119

RMSE Root Mean Square Error. 12, 42, 49, 65, 69–71

RoboGEM Robot-Centric Group Estimation Model. 10, 33, 34, 59, 71

RoSAS Robotic Social Attributes Scale. 48

SALSA Synergetic sociAL Scene Analysis. 10, 14–18, 24, 32, 36, 42, 49, 59, 84–92, 97, 100, 102, 104–106, 108–111

SANG Socially-Aware Navigation between Groups. 15, 17, 101, 103, 105–113, 116

SII Social Individual Index. 96, 97

SOTA state-of-the-art. 15–17, 27, 28, 41, 42, 44, 46, 50, 64, 68, 72, 73, 76, 80, 84, 89, 93, 95–98, 101–113, 115, 116, 119

SSD MobileNet-SSD. 12, 66, 67, 69–71

TD Temporal Difference. 101

TF Tiny Face. 65–67, 69–72

UAV unmanned aerial vehicle. 44

Chapter 1

Introduction

1.1 Context

Robots are becoming increasingly widespread in our society, and they started to be deployed in densely populated areas. Therefore, in dense, dynamic environments, they need to be able to analyse scenes in a human-aware manner and incorporate this information into their decisions in order to perform their functions in a reliable and socially acceptable manner.

Understanding the social aspects of a scene can enable robots to safely operate in a wide range of environments (e.g, museums, shopping malls) and perform a variety of tasks, such as bringing an object from one location to another, in a socially-aware way. Through Human-Robot Interaction (HRI), social scene analysis enables robots to assist people in performing their tasks, achieving their goals, or helping with their other needs. In surveillance robotics, this type of analysis makes it possible for robots to recognise and prevent abnormal situations or emergencies. In service robotics applications, the ability to analyse a dynamic and social scene allows robots to safely navigate indoor spaces and approach individuals or groups of people.

1.2 Motivation

Service robotics can benefit greatly from social scene analysis and the development of socially-aware solutions. By understanding the social context and mimicking socially-aware behaviour, robots have the potential to fit into and navigate human environments seamlessly. An environment such as a corridor or a large open space only populated by moving human participants has a flow. In general, a flow can be defined as the interactional synchrony of individuals in a group, which results in positive consequences such as more optimal performance and interaction [114]. This phenomenon exists because people rely

on information based on multiple sources, not just a single state of the environment at a point in time. Humans plan and execute their path based on a multitude of factors including but not exclusive to past and predicted future states of the environment, social norms, and their goals, motion, and abilities.

The deployment of mobile social robots in dynamic environments impacts the flow of these settings, as humans often do not have experience navigating the same space with non-human agents. This unfamiliarity is also present in people who have previously been in spaces featuring navigating robots. Users may encounter a variety of robots and they might have different functionality or other programmed behaviour depending on the environment they are deployed in. Consequently, a person will not possess the information required to anticipate the future actions of a robot, which, when applied to all human participants of a given setting, may result in a slowed down or broken flow of an otherwise smooth, dynamic environment. However, if robots exhibit socially-aware behaviour and consequently behave closer to how humans do, human participants in an environment can better anticipate a robot's future actions and plan theirs accordingly due to an increased sense of familiarity. Therefore, programming robots to understand and incorporate social concepts and present socially-aware behaviour in navigation may minimise their impact on the dynamics of a crowded social scene [80].

Taken together, when a robot navigates among people in an environment in a way that is perceived as socially acceptable and familiar, humans will feel more comfortable. According to the work of Kruse et al. [80], this familiarity will result in better anticipation of actions which contributes positively to the flow of the environment. Comfort and familiarity can be measured based on a variety of quantitative and qualitative metrics [47, 101], notably, by measuring a robot's movement similarity compared to humans', its social competence and awareness, and the discomfort it may cause. Movement similarity is considered the extent to which a robot's navigation strategy resembles that of a human. It is not concerned with how individual parts (e.g., wheels, legs) move to create this motion, but focuses on the overall trajectory and movement patters of the robot in relation to human movement. This work presents an approach to socially-aware navigation with the goal of improving familiarity, as that will also improve how well people can anticipate the robot's actions.

1.2.1 Target Setting

An environment can, among other factors important for navigation, be socially characterised by the number of participants in it, the activity they are engaged in, and the social norms exhibited by them. These factors vary significantly between settings depending on whether it's in- or outdoors, the size of the area, and what an environment enables participants to do. Furthermore, they may change based on cultural conventions and otherwise

the demographic of the people in the environment. Therefore, it is crucial to specify the setting and the characteristics of the crowd that a socially aware solution targets.

The work presented in this dissertation focuses on indoor reception or party settings as illustrated in Figure 1.1. The targeted environment is set to have average to high-density crowds according to the definition of Moussaïd et al. [105], or Service level B to D based on the work of Fruin [45]. A crowd is considered a group of people gathered in the same environment. In the targeted setting, people can move around, engage in conversation, form groups, and converge around points of interest, and the crowd's density allows individuals or a robot to navigate the environment comfortably. Participants may also acquire refreshments, and join or emerge from formed conversational groups to engage with their peers. Due to the aforementioned activities, the crowd allows participants to have varied orientations (i.e., people are not facing the direction) and it may include both stationary and moving people at the same time.

In this setting, the robot could be tasked with engaging with people to provide help or information, or with serving refreshments. This setting demands a socially-aware mobile robot due to the nature of the involved tasks, and more importantly, to prevent disrupting the crowd's flow so the robot can be a natural part of the social setting. Such a robot needs to be capable of movement via wheels or limbs, and should not be human-sized in order to minimise risk and damage in case it collides with its environment or users. Moreover, to improve its re-usability and minimise its setup cost in different environments, the robot should only rely on a small range of onboard sensors commonly found on mobile social robots such as a Light Detection and Ranging (LiDAR) sensor and an RGBD camera. With

(a)  (b) 

Fig. 1.1 Sample images from a third-person (a) and a robocentric (b) dataset capturing both individual and group information for conversational group detection research collected in settings where there are people standing either alone or in conversational groups in a reception setting. The images are taken from the (a) Synergetic sociAL Scene Analysis (SALSA) dataset created by Alameda-Pineda et al. [5], and the (b) Robocentric Indoor Crowd Analysis (RICA) dataset which is described in Chapter 3.

the use of non-onboard sensors, there is an additional setup cost, as each setting needs to be equipped with cameras and/or other sensors before an agent can be deployed in it.

1.2.2 Targeted Research Gaps

One of the main challenges in the research area addressing the understanding of human behaviour in a crowded environment is group detection [96]. Group detection is crucial to building a socially-aware system as the way people arrange themselves in groups has an effect on the dynamics of an environment. In this work, a group is defined as two or more people in close proximity who engage in a common activity. Groups might move together, which is an important piece of information a robot can use to better plan its own path through the environment. Moreover, disrupting conversational groups by crossing through them may be considered anti-social [73], diminishing how natural a mobile robot's presence is perceived in an environment. Finally, by identifying conversational groups and recognising their spatial configuration, if a robot is required to join a group in order to provide a service, its ideal position and orientation can be determined in a way that will be appropriate to the social setting [150].

Despite the wide range of potential applications in service robotics, previous state-of-the-art approaches for conversational group detection in crowded spaces seldom involve mobile robots or propose solutions based on sensory information that is not restricted to a robot's onboard sensors. Previous solutions primarily focus on data accessed from a third-person (also referred to as bird-view) perspective [15, 23, 24, 38, 48, 98]. However, capturing third-person data requires the installation of cameras in the environment, which, once fixed, may not be transferable to other settings. On the other hand, a first-person camera provides in situ information and the people the robot interacts with are less likely to be occluded as they can be in the centre of the view. Therefore, a first-person view can provide additional information, and it can be combined with a third-person view when needed. Moreover, previous state-of-the-art solutions for mobile robot navigation in indoor crowded environments [36, 73] have previously not considered groups as social entities, but as dynamic objects. Lastly, based on the work of Gao and Huang [47] and Mirsky et al. [101], the evaluation practices of socially-aware navigation solutions have not been consistent, either.

Based on the aforementioned concepts and gaps in research, this thesis investigates and answers the following research questions:

- How can conversational groups be identified in indoor human populated environments?

- How can group information be used to achieve socially-aware robot navigation in crowded spaces?
- How can social awareness be defined and evaluated in a way that unites the diverse practices in literature in order to allow the reliable comparison of socially-aware robot navigation solutions?

1.3 Contributions

This section details my contributions made to the field of conversational group detection and socially-aware robot navigation. They are outlined in the following list:

- a robocentric dataset which can be used for conversational group detection from a robot's perspective;
- an annotation software enabling the identification of people in a scene and their assignment to identified groups, as well as the labelling of a group's exhibited spatial configuration resulting from the position of group members;
- an unsupervised, Agglomerative Hierarchical Clustering based conversational group detection method capable of identifying groups both from a third-person and a robocentric perspective;
- a supervised conversational group detection method based on a Graph Neural Network embedding with link prediction for the identification of groups;
- a study investigating different sample balancing techniques in order to further improve the detection accuracy of the Graph Neural Network based method mentioned in the point above;
- a Reinforcement Learning based technique for navigating a crowded scene in a real-to-sim setting in a socially-aware manner; and
- an evaluation methodology comprised of a set of quantitative metrics, a qualitative questionnaire, and a Navigation Turing Test [35], previously not employed in a real-to-sim environment.

1.4 Publications

The work in this thesis led to the following publications in peer-reviewed conferences and journals. The list of publications and their brief description is described below.

Schmuck, V., Celiktutan, O., (2020), RICA: Robocentric Indoor Crowd Analysis Dataset, in 'UKRAS20 Conference: "Robots into the real world" Proceedings', pages 63-65.

This extended abstract presents a robocentric crowd analysis dataset specifically recorded for conversational group detection as described in Chapter 3.

Schmuck, V., Sheng, T., Celiktutan, O., (2020), Robocentric Conversational Group Discovery, in '29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)', pages 1288-1293.

This conference paper investigates previous state-of-the-art methods developed for both third-person and robocentric view datasets to detect groups in crowds via unsupervised techniques (as detailed in Sections 2.2.1 and 4.1). It also presents an improved method based on Agglomerative Hierarchical Clustering that relies on multimodal information in order to achieve better detection accuracy by incorporating depth image information. The work in this article corresponds to Chapter 4. In this work, T. Sheng contributed by revising the annotations of the bounding boxes of the dataset presented in the previous contribution.

Schmuck, V., Celiktutan, O., (2021), GROWL: Group Detection With Link Prediction, in '2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)', pages 1-8.

This conference paper presents how social scenes can be represented as graphs and how a Graph Neural Network based embedding can be used in combination with a link prediction technique to detect conversational groups within crowds in a supervised manner as described in detail in Chapter 5. Furthermore, this paper describes multiple supervised techniques including the previous state-of-the-art supervised group detection approach which are detailed in Sections 2.2.2 and 5.1. This work was recognised with a runner-up award in the "NVIDIA CCS Best Student Paper" category.

Schmuck, V., Celiktutan, O., (2022), iGROWL: Improved Group Detection With Link Prediction, in IEEE Transactions on Biometrics, Behavior, and Identity Science (TBIOM), pages 1-12.

This journal article extends the findings of the conversational group detection method first presented in Schmuck and Celiktutan, 2021. Moreover, it investigates sample balancing techniques used in combination with Graph Neural Networks and link prediction in order to improve group detection accuracy. The work in this article corresponds to Chapter 5.

Schmuck, V., Chowdhury, S., Celiktutan, O., (2023), SANG: Socially-Aware Navigation Between Groups, submitted in IEEE Robotics and Automation Letters (RA-L)

This journal article presents a method for creating a navigation algorithm that is capable of more socially-aware navigation compared to a state-of-the-art method. Furthermore, it unifies evaluation practices in the research area of socially-aware navigation by employing a number of quantitative metrics as well as a qualitative questionnaire. This work also presents the first application of a so-called Navigation Turing Test [35] in a real-to-sim crowded navigation evaluation protocol. The work in this article corresponds to Chapter 6. In this paper, S. Chowdhury worked on the initial setup of the simulation environment, its connection to ROS [116], and programming the simulated sensor readings.

Chapter 2

Related Work

There are many approaches focusing on socially-aware navigation that do not necessarily incorporate group detection. Instead, they are focusing on multi-person tracking [67] or automatically learning navigation features through inverse reinforcement learning [141] techniques. However, due to the complexity and diversity of social scenes, these approaches have a hard time learning high-level social norms, such as how a robot should navigate among groups.

To achieve socially-aware robot navigation that incorporates group information, first, the different methods used for conversational group detection need to be investigated. In the following chapters 'conversational groups' will be referred to as 'groups' for brevity.

Group detection involves taking into account the social aspects of how groups are formed in a social setting and the characteristics of such an event in order to derive key features it can be based on. This step also inherently involves the identification of individuals standing alone. Traditionally, we can sort group detection methods into two categories: unsupervised and supervised. Unsupervised techniques have the advantage that they do not require a labelled dataset for training, but only to evaluate their performance. In general, they are usually faster at producing labels than deep learning methods as well. Due to these characteristics, it is worth investigating unsupervised as well as supervised machine learning techniques, as the context of navigation in indoor crowded spaces demands quick decision-making in a real-life setting. On the other hand, supervised group detection techniques can, in general, achieve better accuracy, making the overall scene analysis and navigation framework more reliable. However, supervised techniques demand a labelled dataset to be trained on. This aspect might make them less easy to retrain for vastly different settings such as a space more dynamic than the setting described in Section 1.2.1. An example setting could be one where people and groups need to reach their respective goal locations in an environment such as a mall or an airport. Regardless of which group detection method type is used, in order to evaluate possible solutions group detection

accuracy needs to be measured. Section 2.1 presents datasets created for the development of group detection methods and Section 2.2 contains a summary of these approaches, as well as how they can be evaluated.

After groups have been identified in a crowd, a mobile robot can use this information to navigate through an environment in a socially-aware manner. Previous socially-aware navigation strategies can be grouped into traditional and Deep Reinforcement Learning based approaches. Traditional techniques in general apply domain knowledge, hand-crafted rules, and heuristics in combination with machine learning methods to traverse crowds. The methods involving Deep Reinforcement Learning define rewards - and penalties - to guide robot behaviour and make the robot learn the characteristics of socially-aware navigation by itself. In order to evaluate the navigation methods produced in either way, we need to establish how to measure 'behaving in a socially-aware manner'. Section 2.3 thus presents both types of approaches to socially-aware navigation as well as an investigation of how the performance of such methods can be evaluated with a focus on social awareness.

2.1 Group detection datasets

To understand the main characteristics of a group, we need to define what can be considered a conversational group, or in broader terms, as it was referred to by Kendon [75], an interaction group. Kendon [75] introduced the social concept of F-formations and investigated how gaze, facial expressions, and spatial organisation change and how these features are dependent on the number of people, type of interaction, and the environment during naturally occurring interaction.

An F-formation can be viewed as the combination of three different social spaces: o-space, p-space, and r-space (see Figure 2.1(b)). The o-space is a convex empty space which is defined by the members of an F-formation surrounding it, and which usually doesn't have any other people in it. The p-space is an area surrounding the o-space, enclosing the bodies of the group members, and the r-space is the area outside of the p-space. Following the terminology established by Kendon [75], I consider an interaction group as two or more people in close proximity who engage in a common activity. The F-formations relevant to group detection within crowds are summarised by Marshall et al. [94] and can be seen in Figure 2.1(a).

In order to train and evaluate group detection methods, datasets capturing people forming groups are needed. These datasets might be collected from a third-person or a first-person perspective. However, evaluating methods on data recorded from a first-person point of view is more representative when the aim is to deploy a group detection solution on a mobile robot, only using the robot's sensors. In general, these datasets have

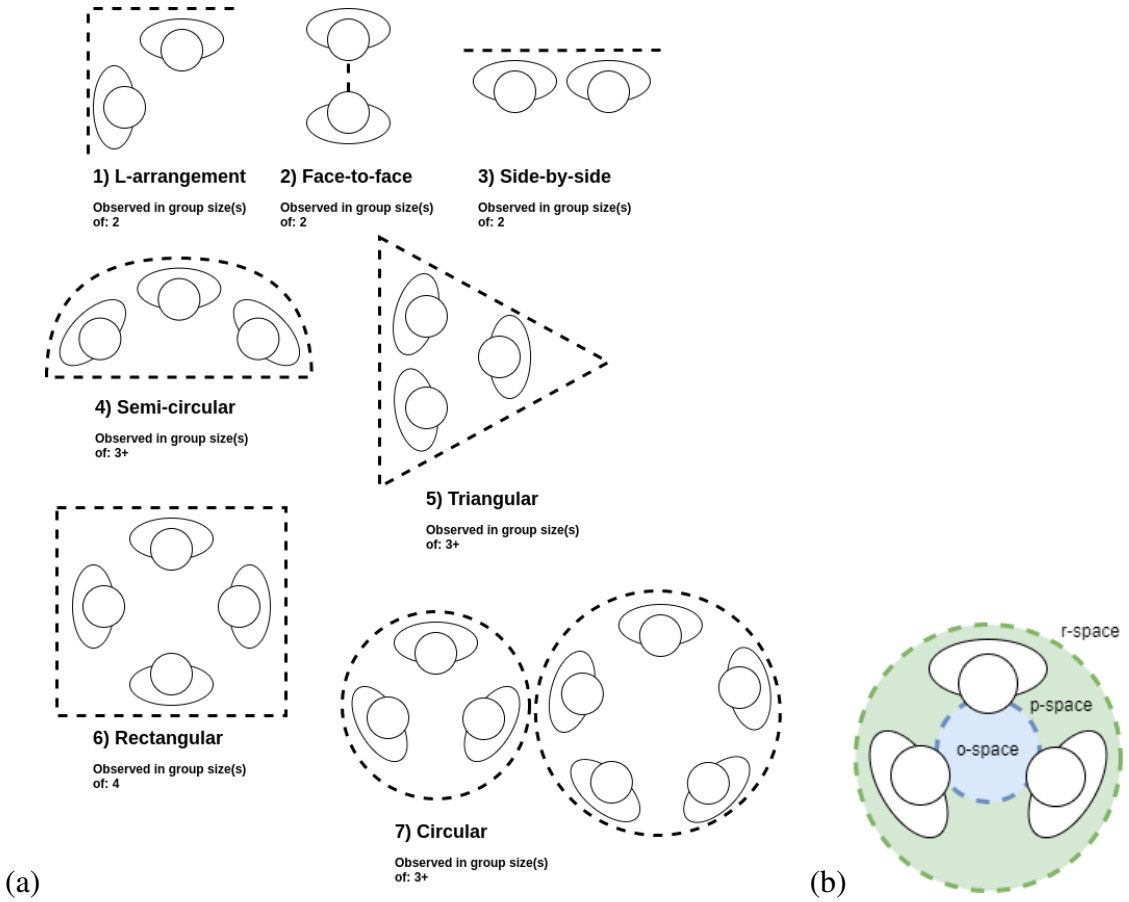


Fig. 2.1 (a) F-formations common in research investigating group detection within crowds; (b) Representation of the o-, p- and r-space in a circular, 3-people F-formation

information on the position and orientation of people, where the position is represented in a 2D top-down mapping of the environment and orientation is the direction in which people are facing. Regarding group information, some datasets only carry information representing whether a person is part of a specific group that was identified, while others also describe the characteristics of a group.

In general, works in this research area explore how individual-level social signals and knowledge about F-formations can be used for the detection of groups in crowded spaces. Using the terminology presented by Vinciarelli et al. [143], the most commonly used social signals include *posture* and *walking* patterns, *gaze behaviour*, *focus of attention*, *distance* from people and objects, *standing* and *seating arrangements*.

Sections 2.1.1 to 2.1.3 describe three datasets collected from a third-person (see Section 2.1.1) and a robocentric (see Sections 2.1.2 and 2.1.3) point of view, which capture multiple groups in an indoor environment and hold information about some of the individual-level elements required for group- and F-formation detection described

(a)  (b) 

Fig. 2.2 Sample images from the Synergetic sociAL Scene Analysis (SALSA) dataset [5]. The images were taken from the 'cam1' footage of the: (a) SALSA Poster session (SALSA-PS); and (b) SALSA Cocktail Party (SALSA-CPP) subsets.

above. For the sake of completeness, Section 2.1.4 presents other datasets connected to F-formation detection.

2.1.1 Synergetic sociAL Scene Analysis dataset

The Synergetic sociAL Scene Analysis (SALSA) dataset was presented by Alameda-Pineda et al. [5] and it was recorded from third-person viewpoints by four RGB cameras. SALSA holds annotated information about groups in two indoor social settings and it has been widely used since its publication.

The two parts of the dataset were captured during a poster session (SALSA-PS) and a cocktail party (SALSA-CPP). The resulting subsets have 627 and 500 images respectively. Examples of the two settings can be seen in Figure 2.2.

SALSA holds over an hour-long recording of data about 18 subjects interacting with each other for approximately 60 minutes at two social events. It captures information about the position, body orientation, and head orientation relative to the body. SALSA is annotated every three seconds with regard to exhibited F-formations, capturing which people form groups, but it does not have information regarding F-formation types. In addition, the Big Five personality questionnaire [51] results of the participants were also recorded. The F-formation annotations were later revised and further validated [154] by taking the agreement between three annotators.

In this dataset, group information is represented by grouping the IDs of participants on each annotated timestamp. The groups are not serialised, but they do keep the same order of appearance throughout the samples. An example of this can be seen in Table 2.1.

SALSA is a relevant dataset to the research presented in this dissertation as it can be used for the training and evaluation of group detection solutions owing to its annotation. Moreover, as it captures two reception-like social environments, it fits the specific setting

Table 2.1 Example group information from a selected timestamp taken from the SALSA-PS subset. Each row represents a group observed at 3.2 s.

Timestamp (s)	Person IDs in a group
3.200000	8, 12, 17, 4
3.200000	2, 13, 7, 3, 6, 5
3.200000	15, 1
3.200000	9, 11, 16, 18, 10, 14

outlined in Section 1.2.1. However, it is a third-person dataset, and therefore, methods developed for group detection which are to be applied to a real robot using only robocentric data should not be solely evaluated on it.

2.1.2 Robot-Centric Group Estimation Model dataset

The shift from a third-person view to a robocentric one introduces different challenges. These include, but are not exclusive to a narrower field of view, dynamic camera, non-ideal illumination conditions, and noise resulting from the perspective of the view. This noise can be in the form of people or static objects occluding people, motion blurred and shaking footage due to the combination of the movement of the robot and not recording with a balanced camera. In order to test solutions meant to be deployed on robocentric systems, they need to be tested on robocentric datasets.

While most datasets which are annotated to accommodate the training and testing of group detection solutions are taken from a third-person view, the Robot-Centric Group Estimation Model (RoboGEM) dataset proposed by Taylor et al [134] was collected from a robocentric perspective. Their dataset was collected in outdoor environments including a public park, sidewalks, and indoors in corridors by a teleoperated robot that was driven around during daytime in settings with varying crowd density. The resulting data contains 16827 RGBD images collected during the course of 1.5 hours. RoboGEM contains annotations for 5423 non-unique groups between frames, and single-annotator bounding box highlights in 14710 images. An example of this annotation can be seen in Figure 2.3. To verify the validity of their annotations, a second annotator also labelled 2000 randomly selected images from the dataset and compared their agreement based on precision, recall and Intersection-over-Union (IoU) scores.

RoboGEM showcases most characteristics of a group detection dataset described in Section 1.2.1 of this work. However, despite having been collected from a robocentric perspective with a mobile robot, and having group annotations, it does not fulfil the criterion regarding the targeted setting. Contrary to the setting described in Section 1.2.1, RoboGEM was primarily collected in outdoor settings, with moving pedestrians in a large

Fig. 2.3 A sample image taken from the Robot-Centric Group Estimation Model (RoboGEM)[134] dataset showing an outdoor daylight setting while the data collecting robot is in motion and pedestrians are moving in front of it. The bounding boxes show example annotations of two two-people groups.

variety of crowd densities, which is not representative of an indoor, crowded reception-style setting. Lastly, at the time of writing, RoboGEM is not publicly available.

2.1.3 JackRabbot- Dataset and Benchmark

Similar to RoboGEM, the JackRabbot Dataset and Benchmark (JRDB) is also a robocentric dataset which was published by Martín-Martín et al. [95]. The dataset was captured from the perspective of a mobile robot navigating around in 33 and 21 different indoor and outdoor settings respectively. The robot was equipped with multiple regular and wide-angle RGB cameras, and 3D laser scanners recording in 360° around the robot as well as an RGBD camera. The resulting dataset has 64 minutes of image and pointcloud data and was annotated to hold information about people in the captured environment by having both 2D and 3D bounding boxes. The annotation has been done at 7.5 Hz based on LiDAR, pointclouds, and RGB stereo images by creating 3D bounding boxes around each pedestrian, where the assigned ID of people was kept consistent throughout a single environment but not between the 54 different settings. To acquire 2D annotations, the 3D annotations were mapped to the 2D space, and in both 3D and 2D cases, the annotations were upsampled to 15 Hz.

However, JRDB does not hold information about group detections in the recorded settings. A new edition of it presented by Ehsanpour et al. [37] titled JRDB-Act addresses this gap by providing a grouping to the people identified in the original JRDB. In this new edition, the authors added group annotations to all 2.4 million 2D bounding boxes by having instructed annotators assign individuals to group IDs and determine on a 3-

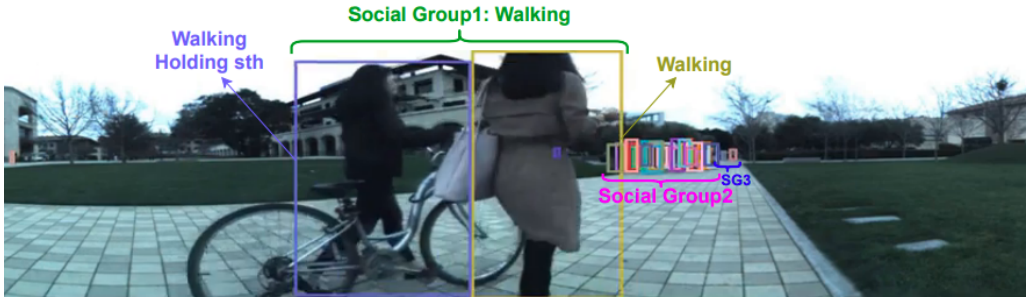


Fig. 2.4 A sample image taken from the work presented by Ehsanpour et al. [37] showcasing an example image from the JRDB-Act dataset. It shows identified individuals grouped into 3 interaction groups, labelled with performed activities, recorded from a robocentric point of view.

point scale how difficult it was to identify the group. The final values of groups were determined by having the majority label assigned by annotators and averaging the difficulty values among them. The groups were also assigned an activity label based on a *pseudo groundtruth* generated from the most frequent action exhibited by individuals who formed a group. An example of this annotation can be seen in Figure 2.4.

The JRDB-Act dataset fulfils the criterion of being a robocentric dataset complete with group and group activity annotations. Its wide range of indoor recording settings make it a challenging dataset for the evaluation of group detection solutions. However, the dataset primarily consists of individuals as 75.5% of the single frame occurrences of individuals cannot be sorted into groups. Moreover, JRDB-Act seldom captures groups consisting of more than four people. According to the reported analysis of Ehsanpour et al. [37], the distribution of social group size is 1.2% for four people groups, and groups of more than four in size appear roughly in the same quantity.

2.1.4 Other relevant datasets

In this section, I present two datasets which were created for investigating groups and group dynamics. They cannot be used for training or evaluating group detection methods in an indoor reception setting, and neither of them is a robocentric dataset. However, they showcase what features are important for the closer analysis of groups, once they have been identified.

The CongreG8 dataset presented by Yang et al. [150] consists of full-body motion-captured third-person data of participants having a conversation in a single group, and being approached by another participant. Each approach takes 2–6 seconds and is performed by either a human or a robot participant, producing 380 and 38 captured approaches respectively. Moreover, participants were tasked with evaluating the politeness of the approaching participant and evaluating their perception of the robot based on aspects

such as politeness and sociability. After the activity, participants were asked to fill out a Godspeed Questionnaire Series[13] which measured the social characteristics of the robot. The resulting dataset presents the social capabilities of the robot with regard to anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety. Taken together, the CongreG8 dataset holds information which can be used to perform approach behaviour analysis, behaviour recognition and generation, and also personality interpretation.

Another third-person dataset titled Babble was introduced by Hedayati et al. [60]. Babble features a third-person recording of seven people interacting in groups of varying sizes. Just like CongreG8, the dataset was captured with motion capture technology from multiple angles where the position and orientation of participants were tracked throughout 35 minutes. This dataset is different from the aforementioned ones as it has labels assigned by two annotators that don't only describe which people's IDs belong to certain groups, but also what the F-formation types of the groups are.

While both of these datasets introduce new concepts such as measuring the social perception of a robot with questionnaires [150], labelling F-formations [60], they were both collected from a third-person perspective. Moreover, both are set in tailored, strictly controlled environments and they don't capture crowds or multiple groups present at the same time.

2.1.5 Limitations and challenges

Based on the practices observed in the works above, group detection in crowds has been investigated before. However, none of the existing datasets can be fully applied to the problem identified in Section 1.2.2. According to my literature research and the findings of Taylor et al.[134], there are several crowd analysis datasets that do not hold annotated information about groups. Moreover, several datasets were collected from a third-person perspective [5, 60, 84, 113, 150], which eliminates some of the noise factors a robocentric dataset would introduce, making group detection less challenging. Moreover, the ones collected from a robocentric view [37, 95, 134] do not hold information regarding F-formations [75] which could be used for achieving better socially-aware navigation and group-approach manoeuvres as described by Yang et al. [150]. Datasets which capture both stationary conversational groups, as well as dynamic participants in indoor social settings, are rare. Namely, only the Synergetic sociAL Scene Analysis (SALSA) dataset [5] and to a limited extent the JackRabbot Dataset and Benchmark (JRDB) [95] satisfies this criterion.

2.2 Crowd analysis methods for group detection

A social setting featuring a crowd can be analysed on two levels. First, individuals can be identified, which establishes the members of a crowd in a social scene. Second, on the group level, identified individuals can be assigned to groups based on who they interact with, following the definition of Kendon [75]. In the following chapters, the identification of individuals in a crowd was considered as 'person-level', while the grouping of the identified people is termed a 'group-level' analysis of a scene.

The following sections present the relevant literature about unsupervised (see Section 2.2.1) and supervised learning based (see Section 2.2.2) methods for group detection. The methods used to evaluate these approaches are described in Section 2.2.4.

2.2.1 Unsupervised group detection methods

One of the observed approaches used in connection to group-level scene analysis involves the approximation of the people's visual fields. In this line of work, Bazzani et al. [16] investigated the effect of using faces instead of full-body posture estimation. They estimated head orientations, based on which they approximated the visual field of people in 3D, and used the computed fields of view to create an Inter-Relation Pattern Matrix (IRPM), which showed which people may be in an interaction group in a scene. They computed the IRPM by taking two individuals at a moment in time and checking whether their respective approximated visual fields satisfied three criteria. The people need to be closer than 2 meters, which is the upper limit of the so-called *social consultive zone* [144], their computed visual fields need to overlap with each other, and their heads need to be inside the reciprocal visual fields (i.e., both of them need to see the other's head). If these criteria were satisfied, they were considered to be in the same group.

Similar to this approach, Fathi et al. [41] used face detection to approximate the line of sight of individuals in a crowd. They used the position of the face and the estimated line of sight in order to determine at which point each individual is looking in a top-down grid representation, which they defined as a 'unary' term. Moreover, they factored in the fact that if someone is looking at a certain location in a scene, the chance of someone else also looking there is higher, which was defined as a 'pairwise' term. For group detection, they first estimated people's attention in a scene with its unary terms, creating an initial grouping of individuals. Then in the second stage, their algorithm considers both unary and pairwise terms to merge or split groups, creating a final group representation.

Following the main ideas presented by Bazzani et al. [16] and Fathi et al. [41], multiple works investigated the use of pairwise affinity matrices in connection to the problem of group detection. Vascon et al. [140] proposed an approach based on the position

and orientation of people, which embedded constraints defined by F-formations into a game-theoretic probabilistic approach. They calculated the attention frustums of people and modelled their pairwise relations before using the matrix generated based on this information in a non-cooperative game to produce clusters. Inaba et al. [69] also proposed the calculation of affinity matrices; however, instead of detecting F-formations, they treated groups as Dominant Sets [112], representing individuals as a graph created based on their position and orientation. To detect groups, they defined a "standard quadratic optimisation problem" that they solved with a method taken from evolutionary game theory called replicator dynamics.

In a different direction, Cristani et al. [31] proposed Hough Voting for F-formations (HVFF), which took positions and head orientations of identified individuals. Moreover, they established constraints such as two individuals forming a group need to be at maximum 1.5 meters from each other. Based on these features and constraints, they employed Hough Voting to identify F-formations, specifically the centres of the o-space of an F-formation, and assign individuals to them. Building on this line of work, Setti et al. [124] proposed a multi-scale extension of Hough Voting based on Graph Cuts, GCFF. They improved HVFF [31] by generating multiple possible o-space centres within a possible space at a set transactional distance away from an individual. Then, similar to a k-means algorithm [91], the cost of an objective function is minimised via a graph-cut based optimisation [81] that assigns people to o-space centres until the algorithm converges.

Furthermore, Japar et al. [70] proposed a method based on single image data. They first detected faces with the TinyFace detector [64], then used bounding box corners and centroid coordinates as feature vectors with an array of linkage algorithms to perform Agglomerative Hierarchical Clustering (AHC) [48]. Due to the unknown number of groups which need to be identified in a scene, they calculated the Davies-Bouldin index [34] for each possible number of groups (K) ranging from 1 to the number of detected clusters, and then selected the K value giving the lowest score to determine the optimal number of groups in an unsupervised manner. Similarly, Taylor et al. [134] proposed a hierarchical clustering based solution. They also used bounding box and depth information of people and they added optical flow vector features computed based on pedestrian motion estimation. Their resulting feature vector was used to perform hierarchical clustering to detect groups.

Based on the literature presented in this section, unsupervised methods can be applied to the problem of group detection in crowded scene analysis. Most works focused on taking body and/or head information after performing a person-level analysis of a social scene. Regarding the approach, a trend can be observed where methods investigated pairs of people in a scene and aimed to use this pairwise information to detect groups, often creating pairwise affinity matrices. It can also be observed that unsupervised group

detection can be done without the explicit computation of gaze direction and field of view, using only raw orientation and position information instead.

2.2.2 Supervised group detection methods

While unsupervised methods do not require training data and therefore annotations, they might also lead to lower group detection accuracy. Accuracy is important as the use of incorrect group information may result in a navigating robot crossing a group's o-space, disrupting the group and thus failing to behave in a socially compliant manner. In the past decade, researchers investigated how Neural Networks can be used for group detection to achieve high accuracy via supervised methods.

Alletto et al. [6, 7] used position and orientation features of people captured in scenes recorded from a first-person perspective. Specifically, they used pairwise relationships between individuals, using their distance and computing the rotation needed for each individual to look at the other person. Using these two features observed throughout multiple frames (temporal window), they estimated which people belong to the same group by using a correlation clustering algorithm [10] in combination with a structural SVM [138] to compute pairwise affinities and iteratively group clusters based on the highest affinity as proposed by Finley et al.[44].

Building on the unsupervised approach of Inaba et al. [69], Zhang and Hung [153, 154] proposed a similar solution. Their method does not work with Dominant Sets [112] based group detection only. After it detects F-formations, it models the social involvement of individuals based on their proximity, orientation and group size. This is done by calculating the relation of individuals' features to those calculated from entire groups they might be associated with. Finally, they modelled the variation and density of potential F-formations in the space with regard to people's proximity and orientation.

Hedayati et al. [60, 61] proposed an approach called REcognize F-FORmations with Machine learning (REFORM), which is similar to the pairwise affinity matrix based methods mentioned in Section 2.2.1. However, their method treats the problem of F-formation detection as a binary classification problem, i.e., whether two people belong to the same group or not. Their proposed classifiers such as Weighted KNN, Bagged Tree and Logistic Regression are used in combination with a greedy reconstruction method that aggregates pairwise relations in entire scenes to detect groups.

The classifier based approach of Hedayati et al. [60] was also followed by Thompson et al. [135], who instead of simple classifiers proposed the use of a Graph Neural Network. They proposed to represent a crowd as a fully-connected graph, mapping individuals to nodes, where the predicted node labels correspond to group IDs. Just as the previously presented solutions, theirs also used position and orientation features when training and

testing on the Cocktail Party dataset [152], and they investigated the use of position, acceleration, image, and semantic features in the case of the MatchNMingle Dataset [19]. After predicting edge labels, they used the generated pairwise affinities in combination with the Dominant Sets [112] algorithm to aggregate the results into detected groups.

Ramírez et al. [117] approached the problem from a similar direction as Hedayati et al. [60, 61]. Using position and orientation data their solution projects a Gaussian-like function in front of a person and uses this projection to establish the relationship between all pairs of people in the scene. This step is taking into account people looking in the same direction or at the same point of interest, or only the position of people in cases when orientation information cannot be reliably acquired due to the setup of the camera or noise. Moreover, they took into account the relative velocity between the pairs where velocities are considered similar if they do not differ by more than $0.2\,\mathrm{m/s}$. Furthermore, if a relationship between pairs has been confirmed, they proposed a remembrance factor, meaning that even if a pair is not evaluated to be in the same group at a later timestamp, those people are not instantly considered to not belong to the same group. Using these computed pairwise relationships as inputs, they clustered people together into groups, finalizing the grouping by computing an intra-cluster synchrony index [21].

Tan et al. [133] also employed a method for group detection based on position and orientation information, and their LSTM-based approach includes the temporal aspect of previously detected groups when computing the affinity matrix representing the relationships between pairs, similar to the work of Ramírez et al. [117]. Using both spatial and temporal context in the LSTM, the resulting affinity matrix can be used in combination with the Dominant Sets [112] clustering approach to identify maximal cliques in the LSTM-generated affinity matrices.

The presented supervised approaches used for group detection mostly rely on position and orientation input features used for the estimation of pairwise relationships between individuals in a scene. They mostly differ in how the pairwise affinity matrices are generated, and they use simpler grouping algorithms such as Dominant Sets [112], greedy reconstruction algorithms, or clustering methods to compute the final set of groups from them.

2.2.3 Limitations of group detection methods

Based on the above overview of how interaction groups can be detected in crowds, it can be observed that the primary source of information for solving this problem is body and head position and orientation, sound, and accelerometer data. However, sound and accelerometer-measured motion modalities are often noisy and thus unreliable in very crowded scenes. Moreover, the majority of presented approaches work with third-person

data, and they do not apply solutions on mobile support robots, which would require a robocentric solution. Lastly, a vast majority of studies build on a bottom-up approach, namely, detecting individuals first and then using unsupervised algorithms to group them.

2.2.4 Evaluation of group detection methods

This section presents what evaluation methods the works presented above used for measuring the performance of group detection algorithms. Two aspects are taken into account, the metrics used to quantify performance and the dataset which was used for the tests.

Bazzani et al. [16] evaluated their Inter-Relation Pattern Matrix (IRPM) based solution on a third-person dataset, PETS 2009 [42]. Among other metrics proposed by Smith et al. [129], they used false positive and false negative counts as well as tracking success rate as the metrics of their evaluation. Cristani et al. [31] evaluated their solution on both synthetic and real data. Their collected Synthetic data (Synth) consists of situations provided by a psychologist which include both groups and individuals captured by one third-person camera, and real data being the CoffeeBreak [31] dataset presented by them. CoffeeBreak presents a social scenario recorded with two third-person cameras over the course of four days. They manually annotated videos of both datasets and computed precision and recall scores (see F_1-score) as well as the Mantel score [93] for comparing their method to state-of-the-art. Fathi et al. [41] collected a first-person dataset via a head-mounted camera, recording a group of eight people throughout multiple days in an outdoor setting. However, they did not focus on evaluating the group detection aspect of their solution, but the recognition of activities performed by the participants, measured via calculated true and false positive rates. To conduct their evaluation, Alletto et al. [6, 7] proposed the EGO-GROUP first-person dataset. It has three distinct contexts such as a laboratory, a café, and a party environment. To measure the performance of their solution, they chose the error, precision, and recall metrics, as was observed in the previous works since the proposed evaluation of Cristani et al. [31].

In the evaluation approaches above, an emerging trend can be observed. Those who do not only evaluate features which can contribute to more accurate group detections measured success primarily based on scores related to the values represented in a confusion matrix (e.g., precision, recall scores).

This way of evaluating group detection solutions was fully established by Setti et al. [124] who took the approach of Cristani et al. [31] for their evaluation metrics, namely generating a confusion matrix and calculating precision and recall scores. Cristani et al. [31] defined a successful detection of a group as correctly detecting $2/3$ of its members. Consequently, in the case of a two-people group, both members need to be assigned to the detected group. The work of Setti et al. [124] extended this in the following way. They proposed

a tolerance ratio ($T \in [0,1]$), where a group can be considered as correctly detected if at least T of its members are correctly identified, and no more than $1-T$ individuals are incorrectly associated with it. Moreover, they suggested focusing on two values for T, namely $T = 2/3$ and $T = 1$. Instead of a Mantel score [93], they calculate the F_1-score from the arithmetic mean of the precision and recall. Finally, they presented a metric independent of the tolerance (T) by measuring the area under the curve (AUC) in an F_1-score – T graph where $T \in [1/2,1]$ and call it Global Tolerant Matching (GTM) score. They tested their solutions on the Synthetic (Synth) and CoffeeBreak (CB) datasets of Cristani et al. [31], the predecessors of the Synergetic sociAL Scene Analysis (SALSA) dataset titled IDIAP PosterData (IPD) [68] and CocktailParty (CP) [152], and GDet following the work of Bazzani et al. [16].

Following the evaluation approach described above, Vascon et al. [140] tested their solution on the Synth [31], CB [31], IPD [68], CP [152], and GDet [16] sets, and adopted the precision, recall, and F_1-score based performance metric analysis with a tolerance of $T = 2/3$.

The method of Ramírez et al. [117] was evaluated on videos generated with a synthetic data taken from a social force model-based simulation. Furthermore, they were evaluated on the Friends Meet [30] and SALSA [5] real-world datasets. They proposed using Normalized Mutual Information (NMI) [132] and Adjusted Mutual Information (AMI) [145] scores instead of F_1-scores, as they argued that due to differing numbers of detected and ground truth groups the F_1-score is not applicable. Instead, they proposed using pairwise- [142] and cluster F_1-scores [65].

Gedik and Hung [49] conducted an evaluation on their own dataset collected in a pub setting with wearable sensors. Their evaluation's objective was to identify the speaking status of participants rather than group detection, and hence they used an area under the curve (AUC) metric and cross-validation to measure their method's performance.

The remaining works [60, 61, 69, 133, 153, 154] all used the F_1-score-based metric for evaluation, occasionally averaging the measured score among multiple frames and measuring the significance of change compared to state-of-the-art methods with a t-test. Furthermore, they all used the SALSA dataset [153, 154], sometimes in combination with other previously mentioned sets such as the IDIAP PosterData (IPD) [68], or proposing their own datasets like in the case of Hedayati et al. [60, 61], who introduced and evaluated their algorithm on the Babble [60] dataset. An exception from these later works is the evaluation method of Japar et al. [70], who evaluated their solution on the ShanghaiTech test set [156] and instead of calculating F_1-scores, they based their performance on identifying the correct number of groups in a scene, and measuring the difference compared to ground truth by calculating the Mean Average Error (MAE) and Root Mean Square Error (RMSE).

2.3 Navigation in a crowd

Navigation of mobile robots in environments with both static and dynamic obstacles, albeit without the social aspect, has been explored in detail [72]. The following sections present how navigating dynamic environments can be achieved with robocentric data (see Section 2.3.1) and what methods have been proposed for incorporating social awareness into robot navigation (see Section 2.3.2).

In general, socially-aware navigation is recognised as a result of a robot navigating in a human environment in a socially acceptable, safe, and efficient manner. Based on the research of Gao and Huang [47], other terms such as human-aware [80], socially compliant [79], socially acceptable [127], and socially competent [97] navigation are also used to describe this kind of behaviour.

Some of the main challenges in this field include the incorporation of social features into the decision-making of a robot. Moreover, one of the main gaps in this area is the lack of a standard evaluation protocol. Therefore, Section 2.3.3 presents evaluation methods that can potentially be used for benchmarking solutions from a social perspective.

2.3.1 Approaches to dynamic obstacle avoidance

Several fast and reliable solutions have been proposed for navigating dynamic environments. The latest approaches make use of fuzzy logic controllers. As described by Faisal et al. [40], these have four main steps: (1) defining linguistic variables for input and output systems; (2) defining fuzzy set; (3) defining rules of the set; and (4) defuzzification. In fuzzy-logic systems, robot sensor readings serve as inputs or linguistic variables, and the final outputs are commands based on which the movement of a robot can be controlled. In accordance with the main steps of fuzzy logic controllers [40] presented above, in steps 2–3 the sensory data is transformed and a rule set is applied to it in order to produce expected outputs after step 4 [107]. Minguez and Montano [100] defined a nearness diagram model which can be used as the fuzzy logic model in such systems. In their implementation, a robot needed to recognise a set of scenarios based on its sensory readings and execute pre-programmed, static behaviours depending on the identified situation it was placed in. This approach allowed the robot to find a short path to its goal and navigate in wide and narrow areas.

A different approach, proposed by Large et al. [82] predicts future positions of detected obstacles via a so-called non-linear velocity obstacle (NLVO) algorithm [126]. By also using an A* algorithm [58], their robot could avoid high-risk situations and adapt its movement to changes in its environment.

In the domain of neuro-fuzzy control [40], instead of having a hand-crafted rule set, neural networks are used to process data and compute the outputs used for navigation.

Lecun et al. [83] proposed a CNN based neuro-fuzzy controller, which, given input from two forward-facing cameras, could successfully produce correct output steering angles for an autonomous off-road vehicle. They achieved good performance by collecting expert data while navigating through different off-road environments, where the steering angle was only changed when it was necessary.

Moreover, Gandhi et al. [46] trained deep neural networks on collision-involving (negative) and collision-free (positive) real-world examples to achieve aerial dynamic obstacle avoidance without subjecting the UAV to uncontrolled, potentially dangerous environments. According to their findings, this approach effectively eliminates the gap between testing in software simulations and the real world, while establishing a controlled environment and utilising a robot prepared for a collision. As a result, their model was capable of flying a UAV in cluttered and confined spaces with both static and dynamic obstacles.

Milde et al. [99] presented a system, based on RGB camera input, for the identification of situations when avoidance manoeuvres are needed. Their evaluation of the solution showed that it was capable of navigating a cluttered office environment by processing camera images with neural networks. The solution demonstrated the capabilities of neuromorphic hardware, placing focus on its increased speed compared to other state-of-the-art solutions used in neural network based computations.

Furthermore, Nasrinahar and Chuah [107] divided the problem of dynamic navigation into recognition–action sub-problems. Their proposed method creates separate fuzzy-logic controllers for static and dynamic obstacle avoidance. They evaluated their solution on a simulated indoor environment holding both obstacle types, and simulated a robot which could analyse its environment with forward-facing distance sensors. When the robot encountered an obstacle, it first determined its type by tracking changes in registered distances over time, then the relevant fuzzy controller performed the selected pre-programmed avoidance manoeuvre. They found that the robot was capable of navigating to its goal without colliding with obstacles.

The above approach of creating separate models for dynamic and static obstacles was also used by Schmuck and Meredith [123] for danger anticipation. Following the work of Gandhi et al. [46], we collected data of negative samples (collisions) and showcased that training two models separately on static and dynamic obstacles results in a significant improvement of danger recognition accuracy, even if the type of obstacle is not detected.

To address the issue of obstacle recognition in presence of occlusion, Huang et al. [66] proposed a solution by estimating bounding boxes around pointclouds retrieved from LiDAR readings. Since there are cases when obstacles are occluded by each other and only a limited portion of their side is scanned, they used the mean and variance of recorded

laser intensities in a classifier. As a result, they were able to successfully detect a number of different obstacles despite occlusion.

As presented above, past implementations of dynamic obstacle avoidance systems were based on features taken from single frames or from sequences of frames, not taking into account the F-formations presented by groups. The majority used deep learning methods to train systems capable of performing dynamic obstacle avoidance while navigating based on both simulated and real-world environments. They also showcased that learning from mistakes can be beneficial for tackling a dynamic obstacle avoidance regardless of the domain (i.e., indoors, outdoors, flying). Lastly, the concept of handling groups as single obstacles appears in the work of Huang et al. [66], which indicates the direction adopted by the works presented in the following section to achieve better socially-aware navigation.

2.3.2 Deep Reinforcement Learning techniques for socially-aware robot navigation

There have been a number of works achieving socially-aware robot navigation by employing Reinforcement Learning techniques. Vasquez et al. [141] introduced two approaches using Inverse Reinforcement Learning (IRL) in a study comparing multiple navigation approaches and two versions of their proposed IRL approach. They proposed a Max-margin IRL [1] and a Maximum Entropy IRL [158] approach, which are both based on optimising model weights until a policy that is sufficiently similar to the training data is found. Vasquez et al. [141] drew the conclusion that the most important aspects of these algorithms are the formulation of their cost function (i.e., the linear combination of weights), and how their input features are designed.

Instead of using IRL, other works explored Reinforcement Learning (RL) approaches based on a variety of algorithm types, input features, and policy optimisations. Chen et al. [25] proposed a multi-agent socially-aware algorithm based on Deep Reinforcement Learning, which introduced social norms into navigation by penalising passing other members of the environment on the right or left side. Their study verified that the inclusion of simple social norms could improve navigation in both simulated and real-world environments. Do et al. [36] proposed an Asynchronous Advantage Actor-Critic (A3C) learning-based approach, which was tested in a simulated environment. In their work, they highlighted the importance of using both sensor-level (e.g., LiDAR, RGB, and Depth images) and agent-level (e.g., position and motion of nearby people) information collected from the robot's environment. Their work showcased that by taking into account obstacles and human positions, as well as the centre points (centroids) of groups, it is possible to create a reward function which enables the learning of a socially-aware navigation policy. Furthermore, their proposed reward takes into account the mean arrival time to a randomly

selected goal, as most predecessor RL-based navigation algorithms do. The method of Do et al. [36] achieved state-of-the-art accuracy in a simulated crowded environment and they proved that the incorporation of group centroids enables a simulated robot to avoid humans and social interactions, resulting in socially acceptable behaviour. They measured social acceptability based on the social individual index (SII) [137] which measures the physical and psychological safety of people and is calculated based on the position of the robot from individuals.

Instead of using group centroids as social information, Katyal et al. [74] introduced convex hulls, which surround groups in scenes, and are similar to the group centroids used in the work of Do et al. [36]. In their work, they diminished the reward the robot is getting if it breaches the generated boundary that marks the edges of a convex hull. Their simulation did not have interaction groups labelled as ground truth, instead, they utilised a Poisson distribution [27] to randomly assign pedestrians close to each other to groups in a scene. The resulting solution was capable of driving a simulated robot in a way that respects individual comfort zones, groups, and minimally sacrifices navigation performance to do so.

The above-mentioned solutions do take into account the social information of crowds; however, most of them are only thoroughly tested in generated, simulated environments. While Do et al. [36] achieved state-of-the-art navigation performance, their solution only considered the centre of mass (centroid) of groups. Moreover, while Katyal et al. [74] introduced the use of convex hulls, in their environment people were not grouped based on interaction groups as those did not exist in the simulation. Based on the related work, the findings of Vasquez et al. [141] hold true, i.e., that the primary reason behind the good performance of RL based methods lies in how their input features are engineered. While the incorporation of group information as a social feature is a promising avenue, previous works were limited in learning from it either due to an over-simplified approach or because the simulated environments the algorithms were trained on did not inherently hold such information.

Another line of work proposed solutions for socially-aware navigation by incorporating trajectory prediction into their models. Zhao et al. [157] utilised Gaussian functions in order to establish an "asymmetric human comfort space" around pedestrians based on their speed and heading. They used this information to generate and update a dynamic cost map, which can be used with an A* algorithm [58] to make a robot navigate through an environment. While they proved that this approach is better than just using an A* algorithm [58] on its own, their evaluation did not prove its viability in crowded environments, and its social awareness was not exhibited since they tested on single-pedestrian settings only.

Finally, Liu et al. [89] proposed a trajectory estimation based solution which also accounts for interaction between individuals in a scene. They trained a Graph Neural

Network to capture interactions between the members of the environment. Then, they used this information in an attention network, which was used to construct a reward function that penalises the robot for getting close to detected interaction groups. They tested their solution in a simulated, crowded environment as well as in a real-world setting. While they concluded that their solution is making the robot's behaviour more socially aware, by not taking into account the robot's static environment, their success rate was hampered.

The above works further showcased that it is possible to create socially-aware navigation with both RL, IRL, and more simple, trajectory based approaches. The key component in these solutions was the incorporation of reliable social features, especially interaction group information.

2.3.3 Evaluation of socially-aware navigation

Previous works on socially-aware robot navigation used different metrics when evaluating the quality of their solution. In principle, the trends in literature can be grouped into quantitative and qualitative approaches.

Apart from standard navigation descriptors such as measuring the rate of successfully reaching a goal position, time and distance travelled, and computation time, conflict counting is one of the most common metrics when it comes to quantifying social awareness of robots [101]. Conflict counting means the counting of the robot's interaction with its environment that prevented it from safely reaching its goal. Examples of conflicts could be collisions with static obstacles or the boundaries of environments and dynamic objects, or colliding with people or crossing groups, where the collision happens with the imaginary boundary indicated by the o-space of the group [75] (see Figure 2.1(b)). In addition to conflict counting, Mirsky et al. [101] and Gao and Huang [47] both highlighted the following metrics for measuring social awareness and/or human discomfort:

- Movement similarity – Measures how a robot moves in a space with regard to its 2D trajectory compared to how humans do. For instance, when moving from point A to B in a crowded place, how similar is its path taken to a person's;
- Smoothness – Measures path irregularity, which can be quantified by measuring the amount of unnecessary turning over the travelled path;
- Avoidance distance – Measures how close the robot comes to humans. This is a good indicator to see if the robot is passing someone too close for comfort or even colliding with people; and

- Compliance with spatial models for groups – Measures the robot's distance from group members compared to the members' distance within the group, and if the robot is a part of the group, whether it's in the field of view of human members.

Gao and Huang [47] also detailed qualitative metrics, which they state contribute to the psychological safety of the human members of an environment. Qualitative metrics include the measurement of comfort levels over time on a 5-point Likert scale [87]. Other metrics may come from the perceived safety segments of the Godspeed questionnaire [12], and the Robotic Social Attributes Scale (RoSAS) questionnaire [22], which measures the robot's social attributes such as warmth, competence, and discomfort. Lastly, they highlighted the Perceived Social Intelligence (PSI) scales [11], which evaluate social competence, awareness, sensitivity, and perceived social skills. The use of PSI scales was also supported by a study by Honour et al. [63], which found that socially-aware navigation based agents consistently score higher on PSI, making this qualitative measurement indicative of good socially compliant behaviour.

Apart from the above-mentioned quantitative and qualitative measurements, Devlin et al. [35] proposed a Navigation Turing Test (NTT) with the aim of measuring human-like behaviour. They argued that in the case of navigation, human likeness cannot be measured purely based on success rate and quantitative metrics, and that qualitative measurements do not provide a good baseline when participants only analyse footage depicting agent navigation in isolation. This is the first work featuring a comprehensive evaluation in this domain employing the principles of the original Turing test [103], by establishing a protocol for comparing human-controlled to agent-controlled navigation via videos taken from different agents navigating an environment.

2.4 Summary

The literature presented above showcased the progression and current state of literature throughout the past years in the research area of group detection and socially-aware navigation.

Publicly available, indoor robocentric datasets used for the training and evaluation of group detection methods that fit a reception-style setting (see Section 1.2.1) are not common. Several datasets [42, 95] that capture groups have not been annotated for group detection. More explicitly, they have person-level annotation, but lack group-level annotation. Moreover, the ones which hold group-level information are often from third-person view [5, 16, 30, 31, 60, 68, 150, 152]. The ones which are robocentric [37, 134] do not hold substantial recordings of indoor environments. However, as a 'good practice' contributing to increased validity, it can be observed that both third-person and robocentric

datasets holding group-level information labelled their data with multiple annotators either for finalising labels based on voting or for checking already assigned labels.

The presented literature exhibited that group detection can be achieved with both unsupervised and supervised algorithms. In general, it can be observed that regardless of the type of method, most works aim to identify relations between pairs of people, creating pairwise affinity matrices [15, 41, 60, 61, 69, 117, 133, 140, 153, 154]. Those who did not propose the creation of affinity matrices proposed techniques based on clustering [6, 7, 70, 31, 124]. These techniques were evaluated on a multitude of datasets, using several metrics, especially in the case of the first publications in this research area. However, since the work published by Cristani et al. [31] which was refined by Setti et al. [124], the research community, for the majority, have been using F_1-scores as a metric to measure how accurate group detection is. Two notable exceptions from this are the work of Ramírez et al. [117] who evaluated using pairwise and cluster F_1-scores, and the work of Japar et al. [70] who calculated the Mean Average Error (MAE) and Root Mean Square Error (RMSE) of detected group counts respectively. As for the datasets used for evaluation, due to the lack of robocentric datasets annotated on the group level, all presented solutions evaluated based on third-person data, in recent years primarily on the two subsets of the SALSA dataset [5].

Regarding socially-aware navigation methods, the literature only rarely takes into account F-formations. In principle, all presented works rely on visual information in the form of RGB or RGBD images and the majority of the works used deep learning methods. Moreover, the ones with more focus on socially-aware navigation usually relied on Reinforcement Learning (RL) or Inverse Reinforcement Learning (IRL) methods. While some of these methods only introduced a social element in their inputs in the form of proximity to individuals, others incorporated some level of group information in the form of treating groups as convex objects [73] and measuring the distance to group centroids [36, 89]. Based on the work of Mirsky et al. [101] and Gao and Huang [47], the evaluation of these approaches has not been unified in previous works. However, they highlighted that both quantitative and qualitative metrics are valuable. They highlighted quantitative metrics (see Section 2.3.3) based on which the social awareness of a robot can be measured and they presented questionnaires which can be used to measure socially compliant behaviour qualitatively. Lastly, Devlin et al. [35] proposed a Navigation Turing Test (NTT) which can measure a robot navigation's human-likeness by comparing videos generated by an algorithm and a real person navigating an environment.

To address the gaps and limitations presented in this chapter, this work presents a crowd analysis dataset (see Chapter 3). The dataset was captured in an indoor, reception-style event from a robot's first-person perspective and holds both person- and group-level information. This proposed dataset enables the development of robocentric group detec-

tion solutions. Regarding the identified gaps in unsupervised group detection, Chapter 4 presents an Agglomerative Hierarchical Clustering based solution which was developed to be applicable to both third-person and robocentric data. To do so this method improved existing solutions to accommodate the shift from a third-person to a first-person perspective. Improving the observed pairwise affinity matrix based solutions, in Chapter 5 I present a Graph Neural Network based solution that leverages the inherent spatial configuration people create when standing in groups. This method achieved state-of-the-art accuracy when evaluated on both third-person and robocentric datasets and exhibited good generality. Chapter 6 presents how group information can be used for socially-aware navigation. Moreover, it presents an evaluation protocol that offers a unified, thorough way of benchmarking and comparing navigation methods to measure social awareness.

Chapter 3

A Robocentric Dataset for Group Detection

Crowded scene analysis from a mobile robot's (robocentric) perspective has not been thoroughly explored before. While group detection datasets have been widely used for safe training and testing within social scene analysis, they do not account for noise factors such as different and changing illumination conditions, motion blur resulting from the shaking or movement of a mobile robot, and people being occluded by each other or objects due to the height of a robot.

While at the time of writing there existed robocentric datasets holding information about individuals, they were only annotated on the person-level, and thus could not be used for group detection. The contribution presented in this chapter addresses these gaps and the ones presented in Sections 1.2.2 and 2.4.

The proposed dataset named *Robocentric Indoor Crowd Analysis (RICA) dataset* was designed to address all the limitations discovered in the literature. It was collected from a robot's perspective, meaning it is robocentric. Furthermore, the mobile robot platform was driven around at an indoor reception setting (see Section 1.2.1). The collection of a novel dataset was necessary as there were no existing unannotated robocentric social interaction datasets capturing the described setting. The dataset was annotated on both person and group levels by multiple annotators following the practice observed in related works (see Section 2.4). The dataset holds labels regarding F-formations of detected groups, which has only been observed in one third-person dataset, Babble [60]. However, as opposed to Babble [60], the RICA dataset captures multiple groups simultaneously.

The details about the collection (see Section 3.1), annotation (see Section 3.2), and features (see Section 3.3) of the RICA dataset are presented below.

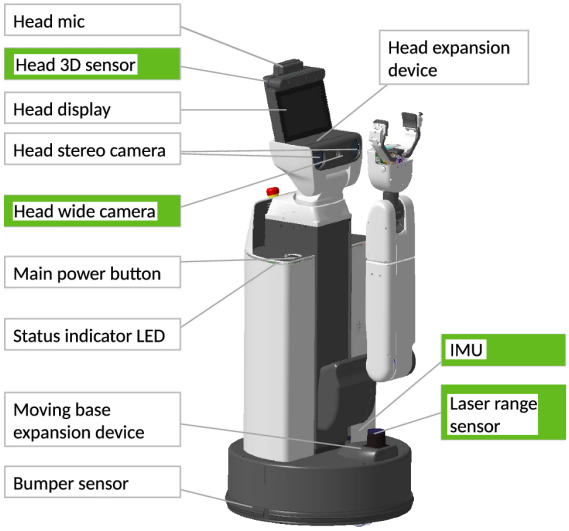


Fig. 3.1 Sensors of the Toyota Human Support Robot series B (HSR-B) [148]. The ones with green backdrops are recorded in the RICA dataset.

3.1 Recording the Robocentric Indoor Crowd Analysis (RICA) dataset

For the recording of the RICA dataset, the Toyota Human Support Robot series B (HSR-B) [148] was used as a robotic platform. HSR-B has 8 degrees of freedom (DoF) for manipulation, 3 DoF of the mobile base, 4 DoF of the arm, and 1 DoF of the torso lift. In addition, it has 2 DoF of its head which has an array of 2D cameras, a 3D (RGBD) camera as well as a microphone for input sensors. Its mobile base is equipped with Inertial Measurement Unit (IMU) and Light Detection and Ranging (LiDAR) sensors. HSR-B has omnidirectional wheels for movement, and its arm is equipped with a gripper, a dedicated 2D camera and a suction cup.

The sensor outline of the robot can be seen in Figure 3.1. The head can be elevated by raising the body of the robot. It can also be tilted and panned, however, this often results in changes in the lighting conditions that are not handled well by the cameras. The remaining relevant sensors are housed in the base of the robot. Its LiDAR can sense in a 240° angle with 0.25° granularity and its bumpers are programmed to perform emergency stops in case the robot collides with an obstacle. In the field of service robotics, various mobile robots are used including quadrupeds like Spot [54], bipeds like Atlas [54], and wheeled robots such as ARI [28], TIAGo [109], Pepper [110], and the HSR-B [148]. Currently, mobile social robots do not typically have anthropomorphic characteristics regarding how they navigate environments, namely, they do not have legs. While this dissertation focuses on creating human-like movement patterns, as described in Section 1.2, this is not regarding how individual robot parts move. Thus, it more practical and cost-effective to focus on

wheeled robots like ARI [28], TIAGo [109], Pepper [110], or the HSR-B [148]. However, ARI and Pepper, lacking a sophisticated end effector for manipulation, may not be suitable as they cannot serve refreshments, which is an intended functionality for the target setting. Therefore, wheeled robots with more advaned capabilities, such as TIAGo and HSR-B, would be best suited for the task. However, the HSR-B has a screen that can be used for displaying information and other Human-Robot Interaction features, making it a more versatile choice for the targeted application.

3.1.1 Recording setup

In order to achieve a frame rate close to 15 FPS regarding the RGB camera recordings of the robot, it was tested to record various combinations of its head-mounted cameras until a desired speed was achieved. Based on these tests, the 'Head 3D sensor' (RGBD camera) and the 'Head wide camera' (wide-angle camera) were chosen (see Figure 3.1) to be captured in the dataset. Their resolution was chosen to be $640x480$ pixels. Moreover, the IMU and LiDAR sensors of the robot were also recorded. The former can be used to match the robot's movement to the captured information of other sensors, and the latter can aid the RGBD camera in distance measurement between the robot and objects or people. Lastly, recording directly to a computer over the local wireless network proved to be detrimental to the frame rate, therefore the robot saved everything onto an attached hard drive. The data was captured via a rosbag [43].

The RICA dataset was recorded during a semi-public departmental event, where around 50 participants were conversing and having refreshments, similar to the SALSA-CPP setting [5]. Prior to recording, the attendees were provided written informed consent. The data collection protocol was approved by the Ethical Committee of King's College London, United Kingdom (Review reference: LRS-19/20-14856).

The environment was an open-plan floor with standing tables and one long refreshment table at one of the edges of the area (see Figure 3.2). To measure crowd density, Fruin [45] proposed 6 levels of crowdedness, based on how many people are in an area, which they refer to as *Level of Service* (see Figure 3.3). In a different measurement, Moussaïd et al. [105] investigated two crowds consisting of different numbers of people and distinguished between a non-crowded environment containing 1098 people in an area, and a moderately crowded one having 3461 in the same area. Calculating the metric by measuring how many pedestrians there are for each m^2 of space ($peds/m^2$), this constituted having $0.03 peds/m^2$ and $0.25 peds/m^2$ respectively in the two settings. According to these metrics, the RICA dataset, having a crowd density of around 22.1 $sq.ft./ped.$ or 0.48 $peds/m^2$, would count as Service Level *C* or more than *moderately crowded* according

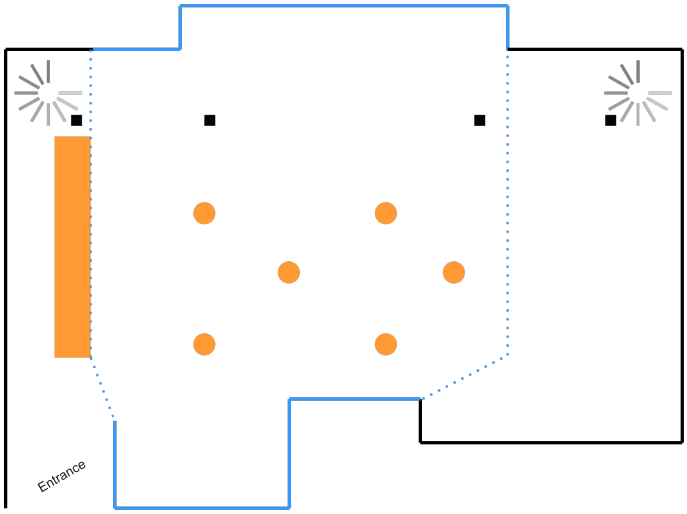


Fig. 3.2 The setting where the RICA dataset was recorded (Anatomy Museum at King's College London, King's Building). The interaction space is highlighted by walls and dotted imaginary boundaries coloured blue. The room features two staircases (upper corners) and four columns (black squares). Moreover, several standing tables (orange circles) and a long refreshment table (orange rectangle) were scattered around the room. The position of the standing tables is a rough estimate.

to the crowd density measurement approaches of Fruin [45] and Moussaïd et al. [105], respectively.

3.1.2 Recording procedure

To obtain a diverse dataset, the robot was driven around at different speeds via teleoperation, following a variety of randomly determined paths. To ensure both individuals' and the robot's safety, the robot was never driven closer than 1.25 m to the participants, and its speed was adjusted according to its surroundings to ensure safety.

To capture a variety of lighting conditions and occlusion settings, the height and head position (i.e., tilt and pan angles) of the robot were varied throughout the capture period. Examples of the camera's captured images can be seen in Figure 3.4. The IMU measurements of the robot and the joint positions of its head were captured while moving, which can be used to find correspondence between image inputs and LiDAR readings.

The RICA dataset was collected over the span of about 65 minutes. The HSR-B hardware was not able to match the RGBD cameras' frame rate observed in other datasets (e.g., in JRDB), but it succeeded in recording LiDAR data at a desired frame rate. The only sensor falling short with regard to the recorded frame count is the wide-angle camera, which only achieved 4 FPS.

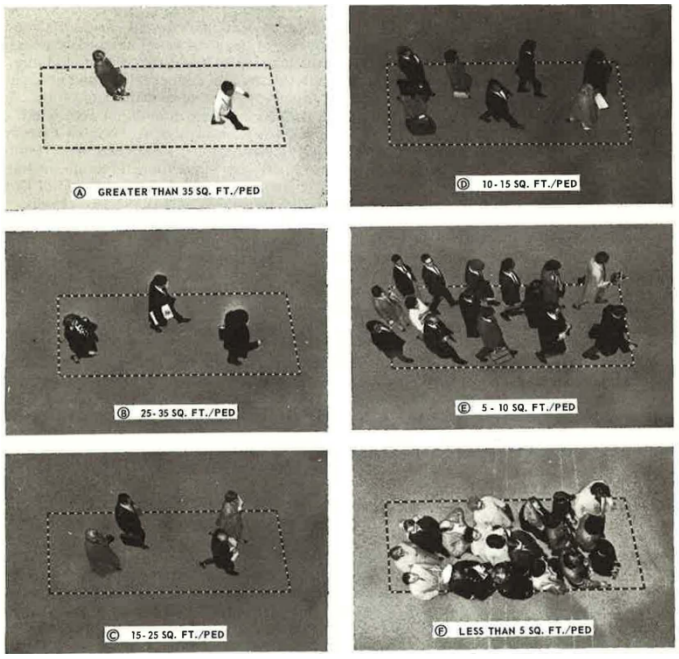


Fig. 3.3 Levels of service from Service level A to F taken from the work of Fruin [45]. The levels display increasing density in a crowd moving from left to right, measured in relation to the area highlighted by the rectangle.

3.2 Processing the RICA dataset

To make the RICA dataset ready for training and testing group detection solutions, the captured data had to be labelled on both a person and a group level. Person-level labelling means the identification of individuals in a crowded scene, and group-level labelling entails the assignment of individuals to groups. Moreover, on a group level, the F-formation [75] of groups can also be identified. The following sections (Sections 3.2.1 to 3.3) detail how these annotation steps were performed and validated to produce a robocentric group detection dataset for crowd analysis.

Not all captured data was annotated, only the first quarter of the entire corpus. Moreover, segments with no people captured for more than 30 s were discarded. This time window was set as it could occur that the robot pans the camera and brings a segment of the environment into view which is empty of people, but due to movement, previously captured people would drift back into the frame shortly after. By not discarding brief empty segments the dataset could also be used for the tracking and reidentification of individuals from a robocentric perspective.

Fig. 3.4 RGBD (a1, b1) and Wide-angle camera (a2, b2) samples from two randomly selected timestamps (a1, a2 and b1, b2) of the RICA dataset.

3.2.1 Actanno-v3 annotation tool

The dataset was annotated with a modified version of the Actanno annotation tool [147]. I upgraded Actanno to use more recent packages, revised and improved its usability so other annotators can use it, and altered it to fit the task of group-level annotation better.

Actanno is short for activity annotation, and the software enables users to view images and draw bounding boxes around their different areas. The software implements hotkeys, bounding box editing functionalities and basic box propagation tools. In this case, box propagation means the copying of a single or multiple bounding box(es) from one frame onto the next one in a sequence. The created bounding boxes can be annotated with a defined set of labels.

Actanno-v3 is a fork[1] created from the Actanno-2 repository[2], which added PNG image compatibility and additional shortcuts among other utilities to the original codebase created in 2016-17. To modify the code, first, its packages were updated to recent releases, syntax errors were cleared and it had to be refactored in a more object-oriented way. A screenshot of the resulting annotation tool can be seen in Figure 3.5.

[1]The Actanno-v3 repository can be found at [122].
[2]The Actanno-2 repository can be found at [53].

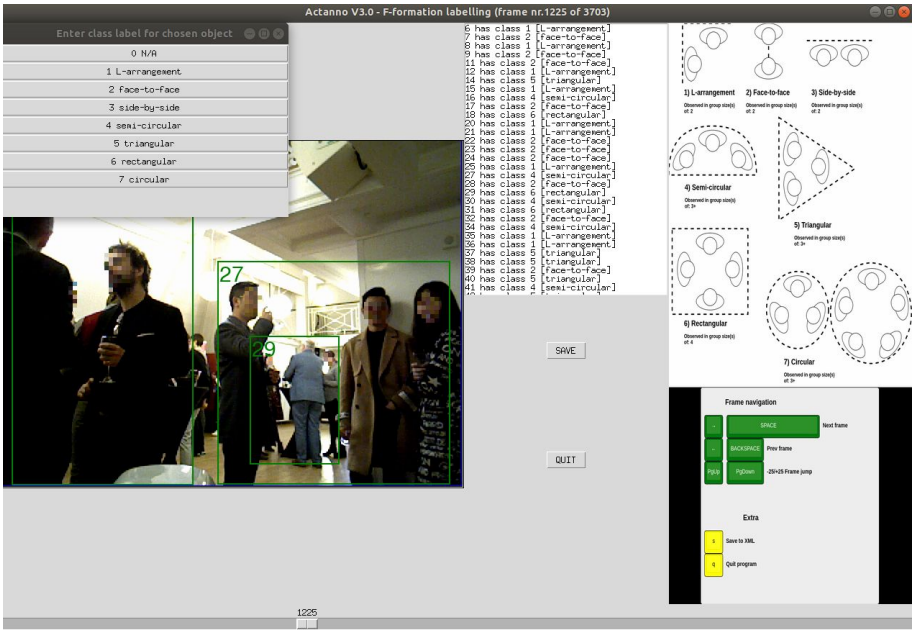


Fig. 3.5 A screenshot of group-level annotation with Actanno-v3. Annotators have visual aids for hotkeys and F-formation types on the right-hand side. They can see the currently annotated image on the left, and a complete list of group IDs in the middle. By clicking a group ID in the list, they can assign a label via the pop-up F-formation selection box (upper-left corner).

One of the main changes I introduced is the split between per-database and per-frame labelling, which was not present in previous editions. This means that while previously only a single label could be assigned to a bounding box, the new version also enables per-frame annotation. This means that a single bounding box can have different labels in each frame, which is essential in a changing environment as it allows a group to change its F-formation label in case a person leaves or joins. The new version also enables index-based annotation as well as the default label set based one. Consequently, a box ID can be associated with a previously annotated parent box. For example, a body part can be assigned to a person with an ID, and an individual can be assigned to a group with an ID. The addition of per-frame labelling introduced very high memory usage, which was reduced by setting up MySQL-based temporary database operations. Moreover, the output naming conventions were unified and improved, new shortcuts were added to help annotators, and user-experience elements, such as pop-up windows informing about successfully executed actions and other visual aids (e.g., shortcut hotkey map) were implemented.

3.2.2 Human and group identification

The dataset was annotated on two levels: 1) on the group level, where the groups were indicated by bounding boxes and were assigned a label based on their F-formation [94]

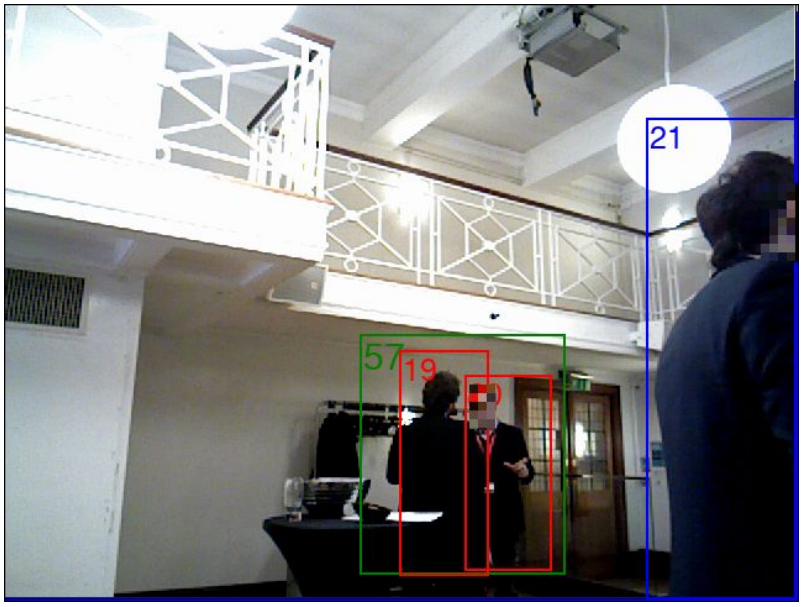


Fig. 3.6 An annotated image recorded with the RGBD camera, showing a person (ID 21 – blue bounding box on the right-hand side) not belonging to any group, and two individuals (IDs 19-20 – red bounding boxes in the middle) belonging to group ID 57 (green bounding box in the middle), where the group formation of group ID 57 is annotated as *face-to-face*.

and a unique identifier; and 2) at the person level, where individuals were indicated by bounding boxes and were labelled based on which group-level ID they belong to.

The bounding boxes for both group- and person-level annotations were first assigned by a trained annotator and later revised by a trained control annotator to ensure all groups and people were reliably identified. The labels of F-formations on the group-level have been decided based on labels assigned by multiple trained annotators and calculated inter-annotator agreement scores. This process is described in Section 3.3, and it was in accordance with the curation process presented by Zhang et al. [154] on the SALSA dataset [5]. An example of a fully annotated image can be seen in Figure 3.6.

The bounding boxes and IDs of both individuals and groups were kept even if only a small segment of a person or only a part (i.e., at least a single member) of the group was visible in a frame. However, if a person or group fully exited the frame, they were assigned a new ID the next time they entered it. This is especially important for group-formation annotations as otherwise, partial groups would receive incorrect F-formation labels. As an example, if a group is formed by *three* individuals, but the robot's panning motion only picks up *one* or *two* of them, the group would, in reality, still exist as a three-member formation, and the positions and orientations of the people, even if fewer of them are in the frame, would still display characteristics of a three-people group. An illustration of this can be seen in Figure 3.7.

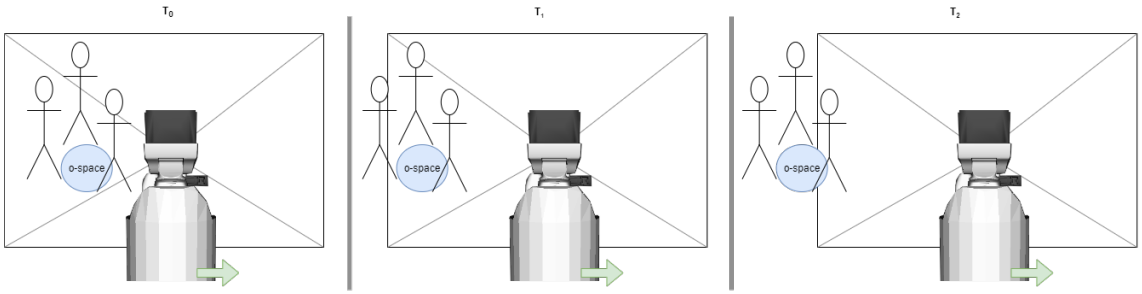


Fig. 3.7 This figure illustrates the importance of labelling groups based on sequence information. The HSR-B's sideways movement is indicated by a green arrow and its field of view by a black rectangle. At T_0 all three people of a group are in the frame, and an annotator would be able to assign a correct label. However, looking at T_1 and T_2 alone, they might think this is only a two-people group or not a group at all. However, in reality, the ground truth of the group label would correspond to that of a three-people one as their position and orientation (not indicated in the figure) reflects the created o-space (blue circle).

3.3 Features of the RICA dataset

The Robocentric Indoor Crowd Analysis (RICA) dataset, published before RoboGEM [134] and JRDB-Act [37], was the first of its kind collected with the aim of filling a gap in the group detection literature. Namely, it features a robocentric indoor corpus which is situated in a reception style setting, just as the widely used third-person dataset, SALSA [5]. Due to the base and head movement of the robot RICA is a challenging dataset featuring noise types such as varying lighting conditions, motion blur, occlusion of individuals by other people, and static obstacles.

The setting was captured with the Toyota Human Support Robot series B (HSR-B) through an RGBD camera, a wide-angle camera, and a LiDAR. Moreover, the robot recorded its joint positions and IMU sensor readings. The RGBDs were captured with a framerate of 10 FPS and can be matched to LiDAR readings for improved distance measurements to detected people and objects.

The corpus consists of over $43,000$ images capturing over an hour of the recorded setting and approximately 50 individuals interacting with each other in groups. Currently, over 10.000 frames of these are annotated on both group- and person-level. This resulted in 194 identified, continuous captures (occurrences) of groups, with occurrence lengths ranging from 3 to 793 frames, with an average length of 146 frames. As for person-level annotations, there are 896 continuous captures of people, their average occurrence length being 85 frames and the lowest and highest being 1 and 1056 frames respectively. 800 instances (89.29%) of human detection are associated with a group in the person-level annotation, leaving 96 occurrences of individuals without a group. There are 122

Table 3.1 Summary of robocentric datasets for group detection.

Dataset	Duration	Participants	Groups	Modality
RICA	71 minutes	50	194	Video, depth, IMU, LiDAR
JRDB	64 minutes	Up to 260 per setting	-	Video, depth, IMU, LiDAR
JRDB-Act	64 minutes	Up to 260 per setting	98	Video, depth, IMU, LiDAR
Babble	34 minutes	7	1	Video, depth, position and motion data
RoboGEM	1.5 hours	N/A	5423 (not unique)	Video, depth

cases where people are observed to either leave or join a group, on occasion changing its F-formation as a result of its occurrence. Following annotation practices described by Taylor et al. [134] and Zhang et al. [154], the identified person and group bounding boxes were assigned and controlled by two annotators. Similarly, to ensure the validity of assigned F-formation labels, three annotators determined group formations and the final group labels were assigned based on inter-annotator agreement. Table 3.1 summarises the features of robocentric datasets. The table compares RICA with other datasets in terms of duration, number of captured participants and groups, and recorded modalities.

Inter-annotator agreement

In order to improve the validity of the dataset's annotations, following the practices of Taylor et al. [134] and Zhang et al. [154], three trained annotators have been tasked with labelling F-formations of identified groups. The annotators were instructed to make their decision based on the number of people present in groups across image sequences instead of relying on single-frame information only, as not all members of a group can be observed at all times due to occlusion, the movement of the robot base, and the joint-position changes of its head.

The agreement between annotators was acquired by calculating Cohen's Kappa [26]. Based on the mean score of $\kappa = 0.4082$, there was a moderate agreement between the three annotators. The pairwise agreement scores between annotators can be seen in Figure 3.8. Moreover, when the most ambiguous F-formation types, namely *Triangular* and *Rectangular* formations were regarded as *Circular* ones of different group size, the measured agreement increased to $\kappa = 0.5123$. This pooling is possible as the *Triangular*

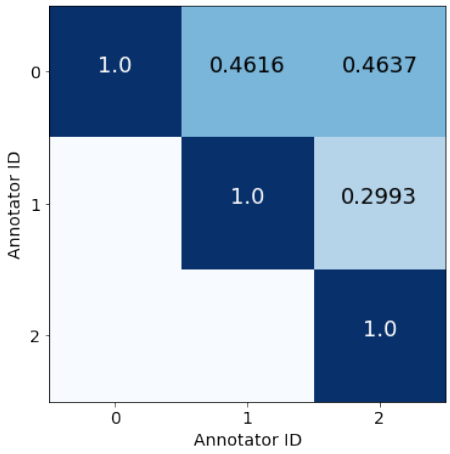


Fig. 3.8 Pairwise annotator agreement without the pooling of ambiguous F-formations.

and *Rectangular* formations are technically inherently *Circular*. Still, due to the lower number of people, the perimeter of the circle formed by them is not as smooth.

To select the correct labels for F-formations, if annotators had a full agreement (68 cases / 35% of F-formations) without converting the aforementioned ambiguous labels, the agreed label was assigned. Similarly, if the labelling presented a $2/3$ agreement over labelling (97 / 50%), the majority vote decided the label. However, in 29 cases (15%), the annotators were not in agreement to any degree. In such cases, the number of maximum detected people across a sequence was used to decide the correct label from the 3 proposed ones. There was a small number of cases (11 / 5%) where this method still resulted in two-way ambiguity, exclusive to the previously mentioned ambiguous labels. Since *Triangular* and *Rectangular* formations can be considered sub-groups of the *Circular* formation, the latter was assigned as the final label for these cases. The final F-formation distribution of the dataset can be seen in Figure 3.10. When the ambiguous labels were considered to be *Circular* formations, the above-described label assignment procedure improved case numbers regarding label agreement to 99 cases (51%) of full agreement, 80 cases (41%) of majority-agreement, and 15 cases (8%) of non-agreement. This shift in the agreement is the reason behind the improved Cohen's Kappa. The above-mentioned protocol for final label assignment can be seen in Figure 3.9.

The above-presented way of handling ambiguity could be improved by involving additional annotators instead of pooling F-formation labels into more generic categories. Although this may be more resource-intensive, its cost can be minimised by having annotators focus only on cases that exhibit a high level of ambiguity. Another approach is to introduce confidence scores to monitor the quality of annotations, as demonstrated in the work of Ehsanpour et al. [37]. Low confidence scores indicate uncertain ground truth labels. These annotations could then either be revised by additional annotators or

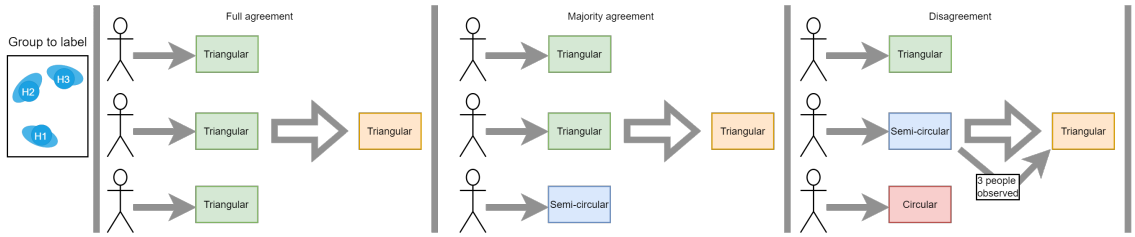


Fig. 3.9 This figure illustrates how the inter-annotator agreement was used to assign final labels of groups' F-formation. The group labelled can be seen on the left, and the final label (*Triangular*) is indicated in an orange box across the other segments. The left-middle segment shows full agreement, where annotators all choose the *Triangular* (green) label, therefore the final label (orange) is also *Triangular*. The right-middle segment shows majority agreement, where two annotators choose the *Triangular* (green) label and one chooses *Semi-circular* (blue), but due to the $2/3$ majority, the label is assigned to be *Triangular* (orange). The right segment shows complete disagreement, where annotators assigned *Triangular* (green), *Semi-circular* (blue), and *Circular* (red) labels and the final label was decided by choosing the one most characteristic of three-people groups, *Triangular* (orange).

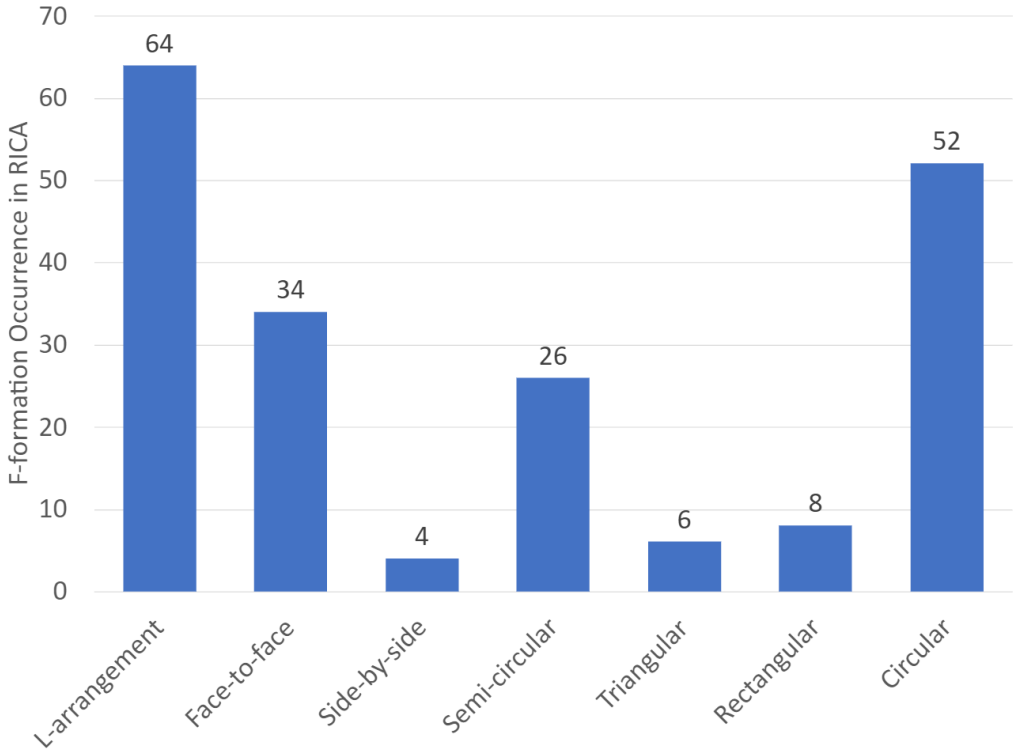


Fig. 3.10 F-formation representation in the RICA dataset across all annotated groups. The bars represent the number of occurrences for each observed F-formation type.

disregarded to improve the overall quality of the dataset. It is important to note, that in the case of the RICA dataset, the number of cases exhibiting a high level of inconsistency between annotators is low, making the cost-effective method described in this Section a viable option for resolving ambiguous labels.

Chapter 4

Clustering Based Robocentric Group Detection

One of the challenges related to socially-aware robot navigation is the detection of groups. This problem can be addressed with both supervised and unsupervised methods. The advantage of unsupervised methods is that they do not need labelled data for training, only for evaluation. Consequently, a reliable, universal unsupervised solution could be deployed in a variety of environments without the need for tuning it on labelled data. Moreover, unsupervised solutions are generally faster to perform inference than supervised ones which are based on deep learning. On the other hand, they are in general less accurate than supervised methods.

As this thesis presented in Section 2.2.1, unsupervised methods have previously been applied to the problem by using the orientation and position information of individuals detected in a scene. However, all investigated solutions were evaluated on third-person datasets, and therefore they might not be applicable to robocentric ones.

The following sections present my contributions to the problem area of unsupervised group detection (see Sections 4.2 to 4.4) and a reiteration of the clustering-based, at the time of development state-of-the-art (SOTA) method (see Section 4.1).

4.1 Analysis of the State of the Art

The state-of-the-art approach for unsupervised group detection in third-person images was published by Japar et al. [70]. Their method was based on single-image inputs and Agglomerative Hierarchical Clustering (AHC) and it was tested on the ShanghaiTech dataset [156].

Due to individuals not being identified in the ShanghaiTech dataset [156], their approach had to incorporate the detection of people. While there are numerous human

detectors capable of identifying bounding boxes of full bodies of people, due to the large level of occlusion characterising the ShanghaiTech dataset [156], Japar et al. [70] used the Tiny Face (TF) [64] detector trained on the Resnet 101 architecture [59] for person-level group detection. However, even using the TF algorithm, they had to exclude too tiny (i.e., smaller than 8×10 pixels) faces and people not facing the camera from their test set. In case of an AHC algorithm, linkage methods define the logic based on which elements in a cluster can be grouped together. Their work presented seven different linkage methods which are:

- Average Linkage – computed based on the average distance between pairs of elements;
- Centroid Linkage – computed by calculating the Euclidean distance between the centroids;
- Complete Linkage – computed by calculating the distance of the farthest elements in two clusters;
- Median Linkage – computed by calculating the weighted centroid of two clusters and similar to Centroid Linkage, measuring their Euclidean distance;
- Single Linkage – the opposite of Complete Linkage, computed by calculating the distance between the closest elements in two clusters;
- Ward Linkage – computed by calculating the best increase in the within-cluster sum of squares if two clusters were joined, where the within-cluster sum of squares is calculated by taking "the sum of squares of the distances between all elements in the cluster and the centroid of the cluster" [70]; and
- Weighted Linkage – computed by recursively measuring distances of potential new clusters compared to other clusters in a set and grouping them if the largest distance is found.

The AHC algorithm of Japar et al. [70] was implemented by providing two feature vectors, one holding information about the position (x, y coordinates of upper-left corner), width (w), and height (h) of a person's bounding box, and another holding its centroid position (c_x and c_y). Since the number of clusters is unknown, they used AHC to generate clusters and the Davies-Bouldin criterion [34] to identify the optimal number of groups. They validated their approach by testing the different linkage algorithms with the two feature vectors and measured the Mean Average Error (MAE) and Root Mean Square Error (RMSE) of detected group counts. They concluded that their approach is capable of

finding coherent group formations in crowds with any of the proposed linkage algorithms, and without knowing the number of clusters in advance.

4.2 Agglomerative Hierarchical Clustering based method

While the method of Japar et al. [70] proved to be applicable to group formation detection, it might not reach the desired accuracy due to the noise introduced by the shift in perspective when a robocentric view is considered. However, to apply this method to the Robocentric Indoor Crowd Analysis (RICA) dataset, the person-level identification of individuals also needs to be investigated (see Section 4.2.1). The pipeline of my proposed approach was as follows. I first obtained the bounding boxes automatically and extracted a set of features describing the location of individuals in a scene. These features were then used as input to the Agglomerative Hierarchical Clustering method (see Section 4.2.3) to find the number of conversational groups in an image. Moreover, adding to the method of Japar et al. [70], I investigated the effect of using a depth modality based on the depth camera recordings of the RICA dataset and feature normalisation (see Section 4.2.2).

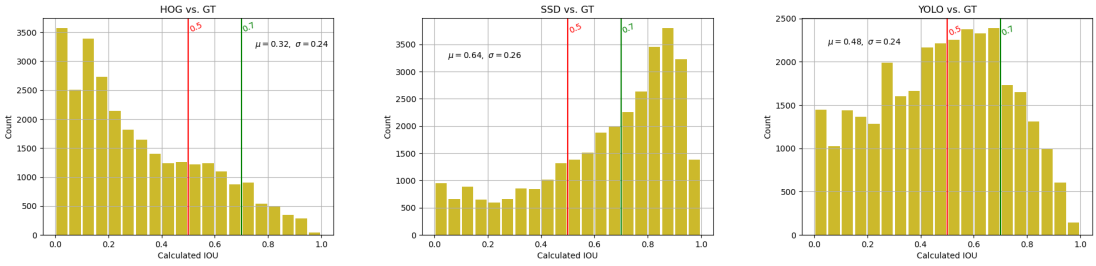
4.2.1 Human detection in the RICA dataset

Since RICA has ground truth (GT) data regarding individuals, it is possible to test different person detectors and the Tiny Face (TF) algorithm proposed by Japar et al. [70] to investigate how well they would be able to detect people in a real-life evaluation where GT is not available.

I tested three methods on the RICA dataset, without fine-tuning: (1) Histogram of Oriented Gradients (HOG) [33] combined with non-maxima suppression (NMS); (2) MobileNet-SSD (SSD) [90] – trained on MS-COCO [88], and then fine-tuned on VOC0712 [104] – with centroid tracking; and (3) YOLOv3[118] – trained on MS-COCO [88]. In addition, I detected faces of individuals with the Tiny Face (TF) algorithm [64] – trained on the WIDER-face dataset [151].

After retrieving the bounding boxes by using all four methods (i.e., HOG, SSD, YOLOv3, and TF), I computed their Intersection-over-Union (IoU) values against GT. However, for TF, since the detected bounding boxes were much smaller than the GT bounding boxes, I computed the ratio between the area overlap between the GT and the whole area of the box detected by TF.

The results of these comparisons are shown in Figure 4.1. The best mean IoU score ($\mu = 0.64, \sigma = 0.26$) was obtained with the SSD detector, and the TF detector yielded boxes with large overlapping areas as compared to GT (area overlap $\mu = 0.88, \sigma =$



(a) (b) (c)

Fig. 4.1 Histograms of Intersection-over-Union (IoU) values measured by the comparison of GT and (a) HOG; (b) SSD; and (c) YOLOv3. The red vertical lines show the minimum IoU and overlap scores to consider a bounding box as a True Positive detection. Green vertical lines indicate the IoU and overlap scores above which the detection is considered as successful.

0.22). Therefore, the outputs of the SSD and TF detectors were used for the detection of individuals in a scene.

4.2.2 Feature extraction from the RICA dataset

Given a single crowd image I, the problem of group detection can be defined as identifying social clusters, denoted by $c = (c_1, ..., c_k, ..., c_K)$ where $k = \{1, ..., K\}$. Differently from Japar et al. [70], in the case of the RICA dataset, there were individuals in the scene who did not belong to any group. K refers to the number of computed clusters as not all clusters correspond to conversational groups. Therefore, conversational groups S can be defined as $S \subseteq c$, where any $c_k \in S$ if $c_k = (c_{k1}, ..., c_{kl}, ..., c_{kL})$ and $L >= 2$, where L is the number of people forming a conversational group.

Once an individual p is detected in a crowd image I using one of the methods described in Section 4.2.1, the individual p can be described with feature vectors A_p and B_p. Inspired by [70], feature vector A_p is defined as $A_p = \{a_p^x, a_p^y, a_p^w, a_p^h\}$, where a_p^x and a_p^y are the spatial coordinates of the upper-left corner, and a_p^w and a_p^h are the width and height values of the bounding box, respectively. Similarly, $B_p = \{b_p^{cx}, b_p^{cy}\}$ is defined by calculating the centroid coordinates (b_p^{cx} and b_p^{cy}) of the bounding box. Japar et al. [70] also reported that the concatenation of the two feature vectors yielded the best results, hence I defined $C_p = \{a_p^x, a_p^y, a_p^w, a_p^h, b_p^{cx}, b_p^{cy}\}$, the concatenation of feature vectors A_p and B_p.

In addition to these 4-dimensional and 2-dimensional features extracted from RGB images, I proposed a new feature using the depth modality. This information can be acquired from the single-channel depth images recorded in the RICA dataset. From a depth image corresponding to an RGB image, I retrieved the depth values within the bounding box for each individual. This is a trivial task as the RGB and depth images have the same

resolution, therefore after matching their sequences based on their timestamps, a detected individual's bounding box information can be used to retrieve their depth values based on its position (x, y coordinates of upper-left corner), width (w), and height (h). This mapping occasionally introduces noise to the depth data due to the noisy readings of the camera itself, and occlusion. As an example, if a person is detected in full body, but an object or another person occludes some parts of their body, the depth readings from the occluded area will reflect the distance of the other person or the object.

To account for noise in the depth data, I computed the weighted average of the depth values instead of taking the raw distance readings. Moreover, to maximise the area occupied by an individual in a box, I used 90% of the detected box areas, resulting in depth values D_{val}. The weight for each pixel D_w can be calculated by:

$$\begin{aligned} d_w^{i,j} &= \sqrt{(b_p^{cx}-i)^2+(b_p^{cy}-j)^2}, \quad \text{for all } i \text{ and } j \\ \text{where} \quad b_p^{cx} - \frac{a_p^w \times 0.9}{2} &\leq i \leq b_p^{cx} + \frac{a_p^w \times 0.9}{2} \quad , \\ \text{and} \quad b_p^{cy} - \frac{a_p^h \times 0.9}{2} &\leq j \leq b_p^{cy} + \frac{a_p^h \times 0.9}{2} \quad . \end{aligned} \tag{4.1}$$

Then, for each individual p, the depth value d_p can be calculated by:

$$\begin{aligned} d_p(D_w, D_{val}) &= \frac{\sum_{n\in j}^{N} \sum_{m\in i}^{M} D_w^{m,n} \times D_{val}^{m,n}}{M \times N} \quad , \\ \text{where} \quad M = a_p^w \times 0.9 &\text{ and } N = a_p^h \times 0.9 \quad . \end{aligned} \tag{4.2}$$

The calculated depth features were combined with the previously presented three RGB image based feature vectors, namely, $A_p^d = \{a_p^x, a_p^y, a_p^w, a_p^h, d_p\}$, $B_p^d = \{b_p^{cx}, b_p^{cy}, d_p\}$, and $C_p^d = \{a_p^x, a_p^y, a_p^w, a_p^h, b_p^{cx}, b_p^{cy}, d_p\}$. Furthermore, to investigate the effect of normalised feature inputs, I calculated the same feature vectors with their respective min-max normalised [4] values, which were denoted by A'_p, B'_p and C'_p for RGB-only features and by A'^d_p, B'^d_p and C'^d_p for multimodal (RGB and depth) features.

4.2.3 Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up approach. Given a crowd image I, each individual p initially represents a cluster with a single element. At each step of the algorithm, two clusters are merged based on similarity, until a single cluster is created. This similarity comparison is guided by different linkage algorithms. In the experiments described below, I used the average [130] and ward [146] linkage algorithms, which were found to be the best-performing methods according to the state-of-the-art results reported by Japar et al. [70]. These linkage methods were presented above, in Section 4.1.

Due to the nature of unsupervised detection, the number of clusters that should be detected (K) is unknown in advance. To determine the optimal number of clusters, two scoring methods can be utilised, the Davies-Bouldin criterion [34] and the Calinski and Harabasz Score [20], which both measure how coherent a cluster is.

The Davies-Bouldin criterion (DB criterion) [34] measures the similarity between clusters by calculating cluster sizes and comparing them to the distances between clusters. The lower the score is, the more separated the clusters are, with the lowest possible score being zero.

The Calinski and Harabasz Score (CH score) [20] is a ratio calculated based on the within- and between-cluster dispersion and it is good at highlighting when clusters are dense and well-separated, which is desirable in group detection. A high CH score [20] indicates clearly distinguished clusters.

The advantage of using the CH score [20] over the DB criterion [34] is that in group detection, especially in dense crowds, groups do not stand far apart. Therefore, an indicator which relies more on the within-cluster distance to measure a detection's cohesion is more beneficial. Therefore, to determine the optimal number of clusters, on each iteration of the AHC method I measured the CH score [20]. The clusters with the highest resulting score were chosen as the solution that best described the image.

4.3 Evaluation procedure

I evaluated the method described in the previous sections for unsupervised group detection on the RICA dataset. In particular, I compared the two methods for obtaining the bounding boxes (i.e., SSD and TF) with the ground truth (GT). I systematically evaluated the proposed three different feature vectors and two different linkage approaches (i.e., average linkage [130], ward linkage [146]) for Agglomerative Hierarchical Clustering (AHC). In Figure 4.2, I present the results in terms of Mean Average Error (MAE) and Root Mean Square Error (RMSE) as proposed by Japar et al. [70] in their evaluation procedure.

It can be observed that the use of average linkage and weighted linkage made no significant difference between the resulting MAE and RMSE scores, as the difference was below 1%. Therefore, I make no distinction between the two linkage types henceforth.

As for human detectors, GT and TF generated bounding boxes both yielded the smallest MAE (1.01) on average, while the generated TF bounding boxes outperformed both other methods based on the RMSE measurements (1.21). This might be due to the fact that bounding boxes are smaller, therefore their weighted average depth information is less likely to include confounding factors such as occlusions in the final feature.

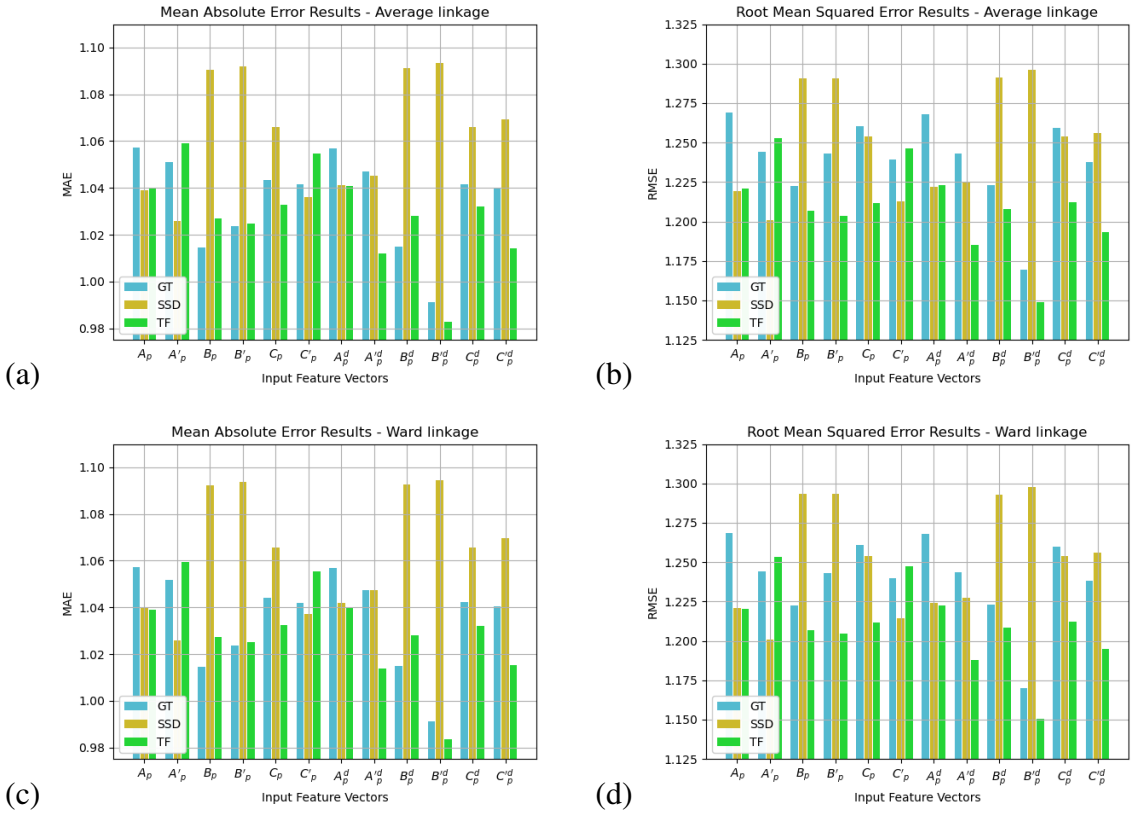


Fig. 4.2 Mean Average Error (MAE) (a, c) and Root Mean Square Error (RMSE) (b, d) results with Average (a, b) and Ward (c, d) linkage methods. $A_p = \{a_p^x, a_p^y, a_p^w, a_p^h\}$: a_p^x and a_p^y – spatial coordinates of the upper-left corner, a_p^w and a_p^h – width and height information of the box. $B_p = \{b_p^{cx}, b_p^{cy}\}$: centroid coordinates ($b_p^{cx}$ and b_p^{cy}) of the bounding box. $C_p = \{a_p^x, a_p^y, a_p^w, a_p^h, b_p^{cx}, b_p^{cy}\}$: concatenated A_p and B_p. $'$-ed terms indicate feature vectors with normalised inputs. d superscripted terms indicate added – d_p – depth feature.

Moreover, it can be observed that the SSD input was outperformed by both other input types in many cases, which can be the result of inaccurate detections (see 4.2.1), as even though it performed best out of the three tested human detectors, its mean IoU score is low.

As for the added depth modality, I present how it improved the error metrics (MAE and RMSE) for 6 feature vectors A_p, B_p, C_p, A'_p, B'_p and C'_p for GT, SSD, and TF generated inputs. As illustrated by the results in Figure 4.2, the added depth modality improves group detection for GT and TF generated inputs as compared to the original features (i.e., A_p, B_p, C_p) proposed by Japar et al. [70].

The best results obtained with GT bounding boxes as inputs are 0.99 (3% improvement) and 1.17 (4% improvement) in terms of MAE and RMSE, respectively, obtained for feature vector $B_p^{'d}$. Similarly, the best error rates for the TF-generated inputs are achieved with feature vector $B_p^{'d}$, and the resulting MAE and RMSE scores are 0.98 (6% improvement) and 1.15 (5% improvement) respectively. When bounding boxes generated by SSD are

used, the added depth information either increases the error rates or has no effect. In other words, adding depth to the feature vectors did not improve the solution in the case of the SSD-generated input. As discussed earlier, this might be due to the fact that the TF-generated boxes are less prone to occlusions when detected correctly, resulting in more reliable depth features.

Lastly, I observed that normalising the input features results in significant improvements when the GT and TF human detectors are used. The presented results show that feature normalisation of this multimodal approach improved the error rate from 1.03 to 0.98 in terms of Mean Average Error and from 1.21 to 1.15 in terms of Root Mean Square Error.

4.4 Discussion

From the person-level detection accuracies, it is clear that one of the main bottlenecks of group-level detection accuracy lies in the performance of the human detector used. This is mainly due to the nature of the robocentric dataset, where people are heavily occluded by each other.

The issue with inaccurate human detections is that a badly detected bounding box will affect most inputs of the feature vector. Introducing noise to depth readings may also happen when using good detections or ground truth information, but the proposed depth feature is calculated by taking the weighted average of depth value readings to compensate for it. Therefore, to support group detections, reliable robocentric human detection methods are needed. Their training and evaluation tailored to a mobile robot's domain could be performed on the RICA and RoboGEM datasets and JRDB-Act.

To address the limitations of the human detector, it may be possible to perform group detection based only on a detected head or other body parts, as observed by the improved performance when using the TF detector. However, due to the noise introduced by changing lighting conditions, some individuals' heads may not be recognised, and even when they are, the detections may not be accurate, as indicated by the average bounding box overlap ($\mu = 0.88$) reported in Section 4.2.1. Detecting other body parts would likely encounter similar limitations. In addition, previous unsupervised works have incorporated orientation information as raw features [16, 31, 69, 124, 140] or by estimating line-of-sight [41, 144]. Orientation cannot be accurately calculated based on single body parts. For example, a single foot's orientation does not give an accurate representation of the body's rotation. While head orientation can be used to estimate line-of-sight, the gaze of individuals wanders during a group interaction. Consequently, raw orientation information only acquired based on the rotation of the head will not be a reliable feature. Therefore, it is more reliable to use full-body information when grouping people if accurate detections can be generated,

particularly if future robocentric group detection research also incorporates orientation information as a feature.

4.5 Conclusion

The previous sections described how an unsupervised group detection solution can be improved to not only work on third-person datasets such as the ShanghaiTech dataset [156] but support a robocentric one. I presented on the Robocentric Indoor Crowd Analysis (RICA) dataset how to enhance and extend a, at the time of development, state-of-the-art unsupervised group detection method by including depth information and feature normalisation. My results showed that both additions improved the overall accuracy of the group detection in challenging robocentric images.

In addition, I compared multiple detectors to acquire human bounding boxes and showed that in most cases detecting faces only (e.g. with the Tiny Face algorithm [64]) could be a better approach rather than taking into account full-body bounding boxes for group detection.

Chapter 5

Graph Neural Network Based Supervised Group Detection

The previous chapters have described that group detection can be achieved with both supervised and unsupervised approaches. While unsupervised techniques have the advantage of eliminating the requirement for a group-level labelled dataset for their training and they are faster to train and perform inference with, supervised approaches are in general more accurate. As presented in Section 2.2, previous supervised group detection approaches were seldom tested on robocentric data, therefore their applicability for deployment on mobile robots has not been proven.

Most previous supervised solutions focused on detecting pairwise connections between individuals [60, 61, 69, 117, 133, 153, 154], creating pairwise affinity matrices and using different grouping techniques to construct groups based on them. This type of approach proved to produce reliable results, but there was a lack of testing on robocentric data and the holistic spatial layout present due to the F-formations created by groups was not incorporated in these methods.

The following sections present my contributions to the problem area of supervised group detection (see Sections 5.3 to 5.6) and a reiteration of the pairwise affinity matrix based, at the time of development state-of-the-art (SOTA) method (see Section 5.1) as well as a background into how Graph Neural Networks work (see Section 5.2).

5.1 Analysis of the State of the Art

The previous state-of-the-art approach for supervised group detection titled REcognize F-FORmations with Machine learning (REFORM) was proposed by Hedayati et al. [60, 61]. Just like other group detection algorithms, REFORM utilises the position and orientation of people. It was created to detect any number of groups of any size based on single frames.

In their approach, they first created a pairwise affinity matrix about each people in a scene, resulting in $n(n-1)/2$ pairs, where n is the number of people in a scene.

The pairs hold information about the position and orientation of individuals, which is used to compute the Euclidean distance and Effort Angle of people. The Euclidean distance (d) between the positions of two individuals (p_1 and p_2) is computed by:

$$d(p_1, p_2) = \sqrt{(p_1 - p_2)^2} \tag{5.1}$$

.

Effort angle is the amount in terms of radians that two individuals would need to turn to face each other. This metric ranges between 0–2π, where the value is 0 if individuals directly face each other, so they do not need to turn to do so, and 2π if they both face away, meaning that they both need to turn π to face one another. Based on these input features, Hedayati et al. [60] trained a Weighted KNN [131], a Bagged Trees [18], and a Logistic Regression [29] method for classification based on these input features of pairs of individuals. The output labels of the classifiers were binary, depicting whether the pair was in the same F-formation [75] or not.

Based on the output of either algorithm, REFORM generated a Relation Matrix, holding information about each pair of people and their classified relation. Based on this relation (or affinity) matrix, they used a greedy reconstruction algorithm based on majority voting. That is to say, if the majority indicates that an individual belongs to an F-formation, then it's considered a part of the group.

They trained their three types of algorithms separately on a randomly selected 60% split of the SALSA-PS dataset and evaluated the resulting models' performance on the remaining 40% as well as on the Babble dataset [60] described in Section 2.1.4. They performed this evaluation protocol, as they wanted to highlight that a single annotated training dataset is sufficient for training REFORM, as afterwards it can be used reliably even on other datasets - in this case Babble [60]. During the evaluation, they measured the F_1-scores as described by Setti et al. [124], comparing their three proposed classifiers, and also evaluating against a benchmark group detection method, Graph Cuts for F-formations (GCFF) [124].

Based on their solution, Hedayati et al. [60] concluded that their REFORM method with the Bagged Tree classifier [18] performed the best regarding F_1-scores, outperforming GCFF [124] and being a generalised solution which can be trained on one training set and reliably tested on a different setting's test set. However, REFORM did not address its applicability to robocentric datasets and based on its training setup, its robustness has not been confirmed, as it was only trained once. As such, its evaluation lacks cross-validation or multiple iterations of random selection for picking its training data.

5.2 Graph Neural Network based methods

As with previously discussed solutions, the applicability of the REFORM method proposed by Hedayati et al. [60] to robocentric input data was not investigated before. Since the target environment of this dissertation was defined as a robot navigating an environment only with its sensors, the method's applicability needs to be verified. Furthermore, REFORM and other, earlier methods were primarily based on pairwise affinity matrices ([69, 117, 133, 153, 154]). However, this problem can greatly benefit from a holistic approach based on Graph Neural Networks beyond pairwise relationships, due to the inherent spatial configuration that exists between individuals who form interaction groups. The sections below demonstrate the effectiveness of a Graph Neural Network (GNN) based approach.

Given a social scene image, the proposed algorithm can use a graph based representation created from position and orientation data to generate feature embeddings for individuals. For robocentric data, I proposed a method for computing the required graph based representation based on multimodal (RGB and depth image) data. These embeddings can be used as input to a binary classifier to determine whether two individuals belong to the same interaction group.

The following sections first provide an overview of GNNs (see Section 5.2.1), GNN based techniques such as representation learning (see Section 5.2.2), link prediction (see Section 5.2.3), and sample balancing techniques which are applicable to GNNs (see Section 5.2.4). Then, my proposed algorithm is presented including how a robocentric view can be transformed into a graph (see Section 5.3.1) and how it was built and trained (see Sections 5.3.2 to 5.4.3).

5.2.1 Graph Neural Networks

Graph Neural Networks have been widely used in a broad range of computer vision tasks ranging from activity recognition based on skeleton data [149], processing facial landmarks [9], and object parsing in scenes [86]. GNNs are constructed from nodes and edges connecting them, where both nodes and edges can have associated features. Typically, node features hold embedded information calculated based on their neighbourhood, defined by their connected edges. Edge features are commonly computed based on the relationship between two nodes, similar to pairwise affinity fields. GNNs have been proven to be better than ordinary Neural Network-based approaches in many computer vision tasks that inherit spatial structures ([86, 9, 149]), and therefore GNNs are a natural means to learn existing relationships between unstructured input features.

In addition, there has been only one work investigating the effect of applying Graph Neural Network (GNN) based solutions in connection to the problem of group detection.

Thompson et al. [135] introduced a GNN based group detection approach, in this case, combined with the Dominant Sets (DS) algorithm [112]. Their solution, similar to my approach described in Section 5.3, also considered individuals as nodes and their relation to each other as edges. However, their final group computation steps differ, as they created pairwise affinities through graph computation layers and computed groups based on DS to exclude from the pool of nodes until they iteratively constructed sub-graphs representing interaction groups. Due to using pairwise affinities only, their solution does not leverage information representation via graph embeddings (see Section 5.2.2), which would allow a more complex knowledge node-neighbourhood representation and they do not use negative injections (see Section 5.2.3) to boost their training samples. Collectively, these two factors lead to a hindered performance.

5.2.2 Representation learning on graphs

Representation learning on graphs allows models not only to gather information from single nodes' raw input features but also to learn aggregated representations based on their neighbourhood. This is achieved by defining a node's embedded set of features based on its raw features or learnt feature embeddings of other nodes which are connected by an edge. Hamilton et al. [57] proposed GraphSAGE, a recent state-of-the-art approach to calculate embeddings for graph node features. GraphSAGE [57] generates embeddings for nodes by learning how to aggregate feature information based on their neighbourhood. This is achieved by employing standard machine learning techniques such as forward- and backpropagation and using stochastic gradient descent. Nodes typically have different numbers of neighbouring nodes and they are unordered, therefore an embedding algorithm is required to be invariant to the size (i.e., number of nodes) and permutation of its inputs. To address this issue, Hamilton et al. [57] investigated mean, LSTM, and pooling based aggregators and have concluded that the mean aggregator, while being marginally less accurate than other methods, is more computationally efficient.

5.2.3 Link prediction with Graph Neural Networks

Link prediction is the process of determining whether two nodes in a graph should be connected by an edge. Zhang and Chen [155] proposed to use link prediction for subgraph detection. Their approach uses graph-embedded information of pairs of nodes as input features to predict whether the nodes should be connected by an edge. Their approach achieved state-of-the-art results and was justified using not only entire graphs but also local subgraphs for learning link prediction models. Moreover, they proposed *negative injection* for training their GNN. Negative injection, while taking existing (positive) edges as

samples, also uses samples of generated node embeddings based on non-existent (negative) edges. This method achieves better generalisation performance as the GNN does not overfit to predict existing edges due to the lack of non-existent edges in the training set.

Contrastive learning [108] can be considered as an alternative to graph embedding generation. Contrastive learning creates embeddings based on individual points' similarity and dissimilarity to others in the same set. While this representation resembles graph embedding at a base level, it still only captures pairwise relations in a neighbourhood. In contrast, graph embeddings with 2 hops as described by [57] (see Figure 5.3) take into account more information. When combined with *negative injection* [155], positive and negative edge-based representations are created for neighbouring nodes, and these features are aggregated to calculate a single node's embedding. This approach generates features that rely on knowledge from the entire set of nodes, resulting in a better representation learnt from the underlying graph structure.

Considering the previous works, it is possible to employ local GraphSAGE [57] embeddings, as Zhang and Chen's work [155] suggests entire network embeddings are unnecessary and computationally complex. Moreover, negative injection should be implemented for better generalisation.

5.2.4 Sample balancing with graph neural networks

While negative injection [155] is a powerful tool used to prevent a link prediction method from overfitting onto only predicting existing edges, it may introduce a different imbalance between positive and negative edges. In a widely used dataset such as SALSA [5], with the employment of negative injection, negative training samples outnumber positive ones roughly with a ratio of 4 to 1.

Previous work has investigated how small adjustments of sample imbalance can be addressed. Abuoda et al. [2] argue that - in line with the findings of Zhang and Chen [155] - adding negative edges to the training samples is a better approach than removing positive ones to counter sample imbalance during link prediction. However, contrary to the negative injection approach, they do not add all possible negative samples to the training pool. Based on this, creating a reliable negative sample pool via negative injection and limiting (e.g., with downsampling) how many of them to use during training might mitigate the sample imbalance problem and further prevent overfitting.

A different approach was proposed by Jiang et al. [71], which used an ensemble learning training technique following the work of Krawczyk [78]. Ensemble learning, similar to k-fold cross-validation, splits a training set into training and validation sets of k folds. Then k models are trained based on their respective folds, and during prediction, all models produce a label. The final prediction is decided by a voting method, which

aggregates the output labels of the different models. While this technique requires more computational time, based on the findings of Jiang et al. [71], it successfully counteracts overfitting without the need to up or downsample the training data.

5.3 Group Detection With Link Prediction

Building upon the previous works on Graph Neural Networks [57, 155], the following sections describe a proposed GNN based group detection approach, called GROup detection With Link prediction (GROWL). Similarly to [31, 60, 61, 69, 124, 140, 153, 154], GROWL makes use of position and orientation features as input; however, differently from these methods, it determines whether a pair of people belong to the same group by taking into account their neighbourhood as well, rather than relying on pairwise affinity fields only. More explicitly, instead of using the deconstructed pairwise representation of a social scene as input, in GROWL people are treated as nodes in a graph. Groups are represented by sets of nodes connected by edges, which are learned by taking a holistic approach.

GROWL is able to detect any number of groups (F-formations [75]) of any cardinality (group size) in crowded scene images captured both from a third-person and a robocentric view. In this regard, it matches the capabilities of Graph Cuts for F-formations (GCFF) [124], the Agglomerative Hierarchical Clustering (AHC) based method of Japar et al. [70], REcognize F-FORmations with Machine learning (REFORM) [60], and the work presented in Chapter 4. As explained below, GROWL is comprised of a graph representation generation and a link prediction step.

5.3.1 Graph representation generation

GROup detection With Link prediction (GROWL) generates graph representations of social scenes by taking into account the position and orientation of people in the image. The position of people is not defined based on the camera angle and distance. Instead, it is a representation of where they are in a scene from a third-person perspective. Since the resulting representation is an undirected graph, this solution is independent of group sizes and changes to the number of detected individuals.

Unlike the SALSA [5] dataset, the RICA dataset is robocentric. To obtain graph representations, the position of captured individuals needs to be mapped from a robocentric view to a top-down one (see Figure 5.1). To compute a top-down representation based on position, the scene can be defined as the area in front of a robot. Based on the robot's RGB and depth camera inputs, individuals can be identified in front of the robot. To determine the position of the individuals (i.e., nodes) along the $x_{topdown}$ (horizontal) axis of the top-down representation, first, the centroid of each individual's bounding box is

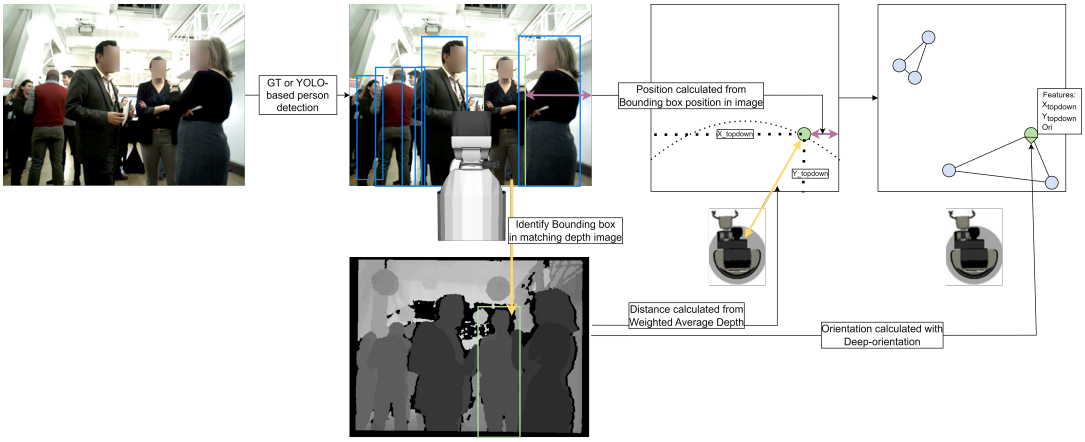


Fig. 5.1 Transformation of a robocentric image (from the RICA dataset) into a top-down representation of people in the scene. An RGB image is used to identify people, and based on their horizontal position within the image calculate their $x_{topdown}$ coordinate. Applying a person's bounding box to the corresponding Depth image, the $y_{topdown}$ coordinate of a person can be calculated by taking the observed depth value, which represents a person's distance from the robot. Finally, the Deep-orientation algorithm of Lewandowski et al [85] can be used to calculate a person's orientation from their Depth information from the robot's perspective. The representation of the robot on the right side is only for reference. Circles represent individuals and are mapped from the robocentric view to the top-down representation. Nodes connected by edges show which people form interaction groups based on ground truth data.

located in the robocentric image, denoted by c_k, where $k = \{1, ..., K\}$ and K is the number of individuals in the scene. Then the x coordinate of c_k is normalised and assigned to $x_{topdown}$. For the $y_{topdown}$ coordinate of the representation, based on c_k in the RGB input, the centroid is mapped onto the depth image to retrieve a single depth reading which can be normalised and assigned to $y_{topdown}$. Once both $x_{topdown}$ and $y_{topdown}$ values are computed, an individual (i.e., node) can be positioned in the top-down graph representation. This representation calculation is illustrated in Figure 5.1.

To compute the orientation of an individual for the robocentric dataset, I used the Deep-orientation framework developed by Lewandowski et al [85]. They investigated lightweight network architectures to perform quick orientation estimation, while staying within reasonable accuracy. They proposed Deep-orientation, a biternion architecture which is based on the structure of the ImageNet VGG architecture [128]. The first stage of their method uses convolutional layers and a max-pooling layer to extract features, and the second stage consists of fully-connected layers with dropouts in-between which determine the angle based on them. The architecture is called biternion as the network has two outputs, which encode the sine and cosine of the computed angle. They tested their solution on different sized input images, also investigating whether RGB, Depth, or RGBD images are the best inputs for their algorithm. They evaluated their solution against the

MobileNet v2 architecture [120] trained and tested on the same data, and they found that while MobileNet v2 was marginally more accurate, Deep-orientation was able to perform almost as well, under significantly shorter time when using Depth images only.

To estimate the orientation of each individual from a robocentric view, I used depth images as inputs for the Deep-orientation network. The raw depth images were segmented based on the bounding boxes of people detected in their corresponding RGB images.

Once the position and orientation features for all individuals in a scene are computed, a graph can be generated based on the graph characteristics described in Section 5.2.2. This is done by setting node features to be the coordinates of people in a top-down scene representation and also adding the orientation of individuals as an additional feature. Edges between two nodes represent whether the individuals belong to the same interaction group and, following the work of Hedayati et al. [60], edge features are computed based on the Effort Angle (see Section 5.1) and Euclidean distance (see Eq. 5.1) of two individuals.

I considered using other features such as LiDAR readings, other camera images or inferred information such as gender or clothing, however, decided against using them for a more fair comparison to the state-of-the-art. Moreover, introducing the use of more features would increase complexity and computing time as well as the need for other sensors to be present on the robot. The overall aim behind these input choices is to achieve on-the-fly group detection with minimal hardware requirement that is comparable to other, similarly minimal designs of previous works.

5.3.2 Link prediction in GROWL

According to the findings of Zhang and Chen [155], selecting training samples for Graph Neural Network based link predictors is non-trivial. A link predictor functions as a binary classifier that predicts whether an edge E exists between two nodes N in a graph $G = (N, E)$, where $E \subseteq N \times N$. However, as mentioned in Section 5.2.3, considering only existing edges (i.e., positive edges) in a graph during training leads to poor generalisation as non-existent edges (i.e., negative edges) will not be represented. Therefore, I perform negative injection by representing the created graphs both in terms of positive and negative edges, where positive edges are denoted by $E_p \subseteq E$, and edges created by negative sampling as $E_n \cap E = \oslash$. A representation of such a graph can be seen in Figure 5.2.

After performing negative injection, each of the created graphs can be denoted by $G' = (N, E \cup E_n)$. In order to represent the information of a node, I use a two-hop GraphSAGE [57] embedding with a mean aggregator. Looking at Figure 5.3, let A be a node from the graph created with negative injection in Figure 5.2. To calculate an embedding for node A, we traverse both positive and negative edges connected to A. Taking the nodes reached this way (collectively h_A^1), we traverse all edges once more from each of them,

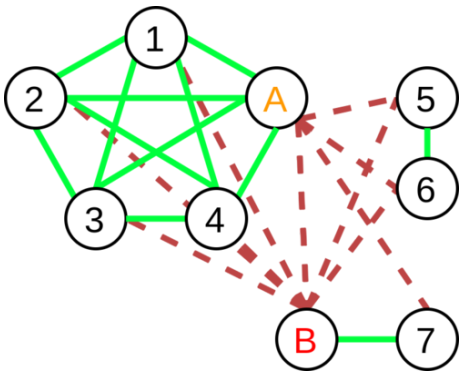


Fig. 5.2 Fully connected graph used for training the GROWL algorithm. Green lines show positive edges and red dashed lines indicate negative edges. The circles with numbers and letters signify individuals in a scene. Based on the green links, the representation shows a five-people circular, and two two-people face-to-face F-formations [75].

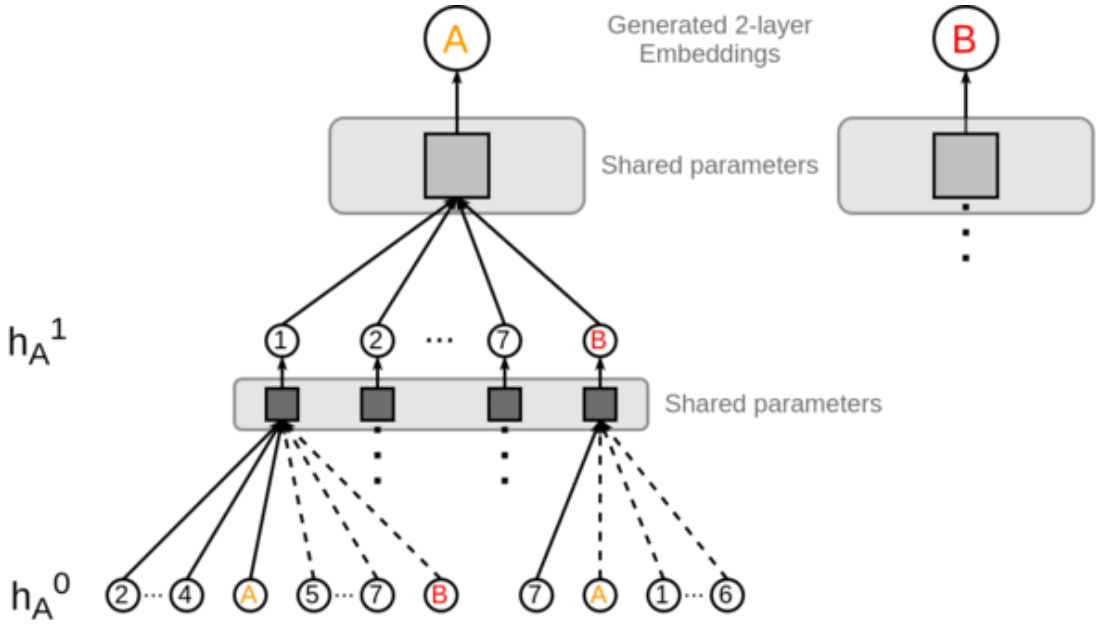


Fig. 5.3 2-hop GraphSAGE [57] embedding network based on the graph presented in Figure 5.2. To get an embedding for node A, we traverse both positive and negative edges connected to A reaching nodes h_A^1. We repeat this step from nodes in h_A^1. Reached nodes are denoted as h_A^0. Values of h_A^0 equal to the position and orientation node features. Embeddings for nodes in h_a^1 are calculated by multiplying the mean-aggregated values of h_A^0 with a weight matrix shared across all nodes on this layer. To get an embedding for node A, another weight matrix is multiplied with the mean-aggregated feature embeddings of h_A^1.

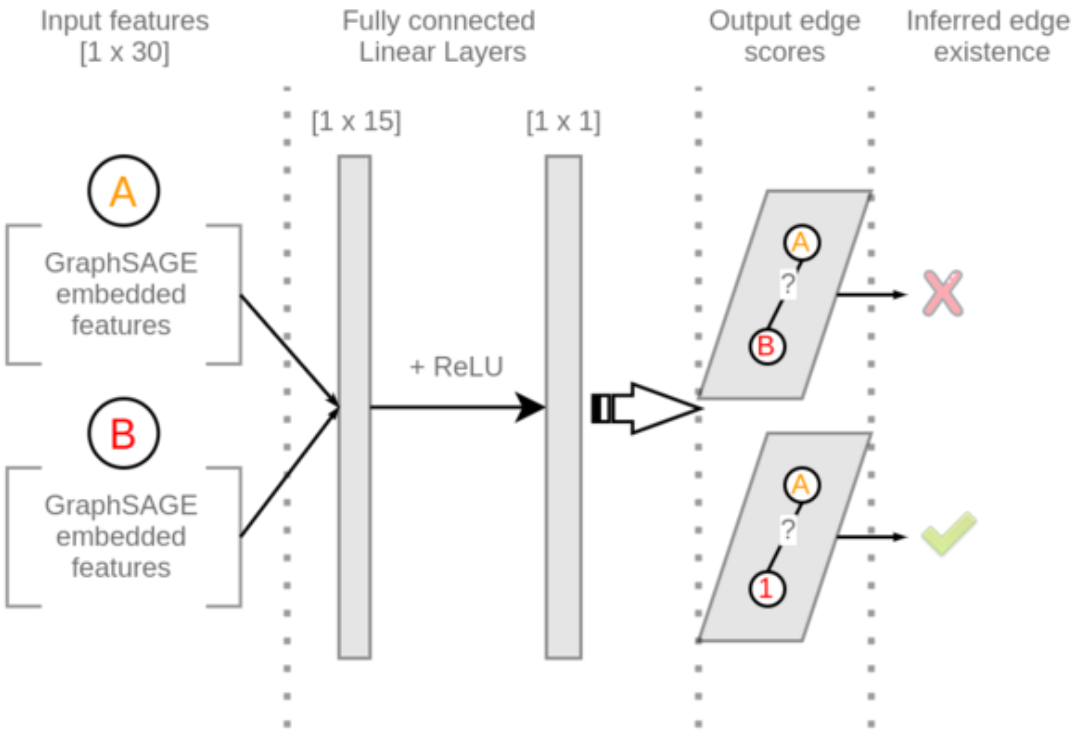


Fig. 5.4 Shallow Multi-Layer Perceptron (MLP) used for predicting existing/non-existent edges in the graph presented in Figure 5.2. The MLP consists of two fully connected layers, with a ReLU activation [3] being applied in-between them. The inputs of the MLP are the embeddings generated by the technique shown in Figure 5.3. The outputs are binary labels signifying whether a link does or does not exist between two nodes, in this case, nodes *A* and *B*.

reaching nodes collectively denoted as h_A^0, making this a so-called two-hop embedding. The values of h_A^0 are equal to the position and orientation node features. Embeddings for each node (e.g., v, u) in h_A^1 can be calculated by multiplying the mean-aggregated values of the previous (h_A^0) layer with a trainable weight matrix that is shared across all nodes on this layer, as proposed by Hamilton et al. [57]:

$$\mathbf{h}_A^1 \leftarrow \sigma\left(\mathbf{W} \cdot \text{MEAN}\left(\left\{\mathbf{h}_A^0\right\}\right) \cup \left\{\mathbf{h}_A^0, \forall u \in N(v)\right\}\right), \tag{5.2}$$

where $\left\{\mathbf{h}_A^0, \forall u \in N(v)\right\}$ is the representation of the nodes in the neighbourhood of node v. σ is a nonlinear activation function, and in the case of GROWL, it was chosen to be the Rectified Linear Unit (ReLU) [3] activation:

$$y = max(0, x) \tag{5.3}$$

Finally, to compute an embedding for node A, once again a different trainable weight matrix is multiplied with the mean-aggregated feature embeddings of layer h_A^1.

During training, calculated embeddings of node pairs are concatenated and passed as inputs for a 2-layer Multi-Layer Perceptron (MLP), which produces a label ($label \in [0,1]$) for each edge. The MLP predictor consists of two fully connected layers, with a ReLU activation [3] being applied in-between them, as shown in Figure 5.4.

The created GraphSAGE embedding model and MLP predictor are trained end-to-end with the Adam optimizer [76] and with binary cross-entropy loss [32] for a number of epochs. To determine the optimal embedded feature size and the number of epochs, I performed hyperparameter optimisation with the SALSA-PS training set described in Section 5.4. The best-performing model was achieved with an embedded feature vector of size 20 if trained for 100 epochs. For a detailed description of the hyperparameter optimisation process, see Section 5.4.3.

Graph pruning

Based on the labels calculated by the MLP predictor, it can be determined if two nodes are connected. However, this information needs to be converted to a graph representation. To do so, a fully connected, non-directed graph is created from all nodes. Then, all edges labelled 0 are removed as illustrated in Figure 5.5, which results in a graph representation of detected groups.

During testing, GROWL, due to negative injection, essentially assumes a fully connected graph and calculates node embeddings for each node via the trained 2-hop GraphSAGE model. The abovementioned MLP predictor can then use pairs of embedded node features to determine which edges are positive (existing) or negative (non-existent).

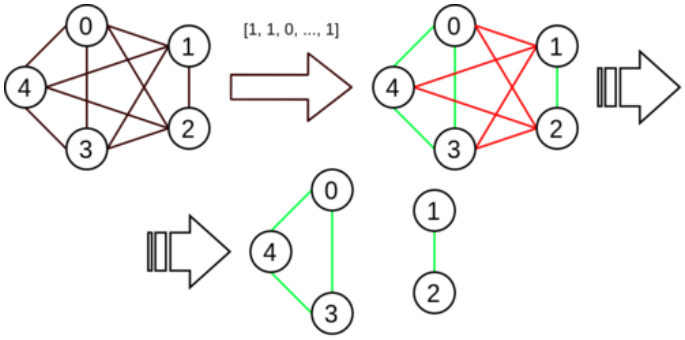


Fig. 5.5 Edge elimination from a fully connected, non-directed graph based on GROWL's output. All edges of a graph are labelled by the MLP predictor; then the ones classified as non-existent ($label = 0$, red) are removed.

5.4 Evaluation procedure

The proposed GROWL algorithm was evaluated on both a third-person view and a robocentric view dataset and the results were compared with a baseline method based on Graph Cuts for F-formations (GCFF) [124] and two current state-of-the-art interaction group detection methods, REFORM [60], and the GNN-based approach of Thompson et al. [135] (T-GNN).

The datasets were chosen to be the Synergetic sociAL Scene Analysis (SALSA) dataset [5] and its subsets, Poster Session (SALSA-PS) and Cocktail Party (SALSA-CPP). In addition, GROWL was tested on the Robocentric Indoor Crowd Analysis (RICA) dataset.

5.4.1 Experimental setup

To examine the generality of the solution, GROWL was trained and optimised on 60% of the generated graphs randomly selected from the Poster Session subset of SALSA (SALSA-PS) [5]. The optimised model was evaluated on the unseen test sets from SALSA and SALSA-PS as well as SALSA's Cocktail Party subset (SALSA-CPP) and the RICA dataset. Hedayati et al. [60] followed a similar approach, namely training on 60% of the SALSA-PS set and testing on the remaining 40% of it.

To compare GROWL with another GNN based solutions, I adopted the evaluation setup of Thompson et al. [135]. They used the ground truth information from the Cocktail Party Subset of SALSA [5] (SALSA-CPP) for training and evaluation of their GNN based solution. They used the first 64 scenes from the dataset for testing, and the last 372 for training their method, which split was adopted for a fair comparison. However, the training hyperparameters calculated during parameter optimisation (see Section 5.4.3) were kept the same, as in all other evaluations.

I would like to highlight that the focus of this work is the detection of interaction groups. Detecting individuals and estimating their orientation in an end-to-end manner is beyond the scope of this work. Therefore, for evaluating the GROWL method, I employed the same procedure as previous works by Setti et al. [124] and Hedayati et al. [60], and I used the ground truth information provided with the Synergetic sociAL Scene Analysis (SALSA) [5] and RICA datasets for a fair comparison, without person-level detection, due to the human detection algorithm bottleneck described in Section 4.4. More explicitly, I used the individuals' ground truth position and orientation information as inputs when evaluating GROWL on the SALSA dataset [5] and its subsets. In the case of the RICA dataset, I only used the ground truth horizontal position of individuals ($x_{topdown}$) and based on their distance calculated their vertical position ($y_{topdown}$) to achieve a top-down representation as described in Section 5.3.1. Orientation was estimated from the RGB and depth data using the method proposed by Lewandowski et al [85], which relies on robocentric depth images (see Section 5.3.1).

To investigate to what degree an automatic human detector has an impact on the accuracy of the GROWL pipeline, I fine-tuned YOLOv4 [17] to detect people in the RICA dataset and evaluated GROWL in two ways with automatically detected bounding boxes. The fine-tuned YOLOv4 [17] had 55% detection accuracy. Fine-tuning was performed on a randomly sampled 20% of RICA, tuning out-of-the-box weights of YOLOv4 [17] based on ground truth human detections for 4000 iterations. When testing the remaining set of the RICA dataset, and during the tests on GROWL, detections were considered successful if the Intersection-over-Union (IoU) value of a detected bounding box compared to ground truth human locations was above 40%.

I evaluated GROWL given the assumption that the automatically detected bounding boxes found by YOLOv4 [17] represent all people in a scene. Due to this assumption, this evaluation (RICA-Y) calculated F_1-scores based on how accurately groups were detected by GROWL compared to ground truth group annotations only including nodes representing automatically detected bounding boxes. Consequently, people from ground truth groups who were not detected by YOLOv4 were excluded. In reality, individuals who are not recognised by human detectors automatically result in False Negative node classifications when calculating group detection Precision and Recall scores, which consequently hinders the final F_1-score of the group detection. Therefore, I also evaluated GROWL with automatically detected bounding box inputs against complete ground truth group annotations.

5.4.2 Evaluation metrics

Calculating F_1-score has been the widely used metric to measure group detection performance. Setti et al. [124] proposed a tolerance ratio ($T \in [0,1]$), where a group can be considered as correctly detected if at least T of its members are correctly identified, and no more than $1-T$ individuals are incorrectly associated with it. In the previous work, T has been commonly set to $T = 2/3$. Based on this ratio, an F_1-score can be obtained for each frame by calculating the precision and recall scores.

5.4.3 Parameter optimisation

The goal of my hyperparameter optimisation was to find the optimal training length in epochs and the optimal feature embedding size of a model. In this work, I searched for parameters where the optimal embedded feature size was within a range between 2 and 20 and an optimal epoch length in the ranges $[10:50]$ with increments of 5 and $[50:250]$ with increments of 50.

To determine which combination of parameters performs best, following the practice regarding how the evaluation was performed (see Section 5.4.1), after randomly shuffling their order the parameter optimisation algorithm randomly selected 60% of generated graphs from the Poster Session subset of SALSA (SALSA-PS) [5]. On this set, a 10-fold cross-validation was performed, resulting in approximately 362 samples in each training set, and 38 samples in validation sets. I measured mean F_1-scores and standard deviation for the validation folds. To ensure the results are not affected by a lucky training-test split, I repeated the above cross-validation procedure 3 times, randomly selecting 60% of the shuffled graphs from the SALSA-PS dataset each time, resulting in 30 validation samples for each hyperparameter pair.

Based on the experiments, the best performing model was achieved with an embedded feature vector of size 20 and an epoch count of 100, which achieved a mean F_1-score of $\overline{F_1} = 97.7\%$ with a standard deviation of $\sigma = 13.1$.

Sample balancing

Based on measurements taken by running multiple evaluation rounds during parameter optimisation (see Section 5.4.3), it could be observed that randomly selected training samples with at least $23.4k$ positive samples and a difference between positive and negative sample count that's smaller than $65.5k$ consistently produced the best results. Motivated by this, I implemented strategies to better balance training samples. Therefore, I investigated the effect of negative sample downsampling following the work of Abuoda et al. [2] and ensemble learning based on the findings of Jiang et al. [71].

I observed that the gap between positive (S_p) and negative sample (S_n) counts ($\Delta NP = S_n - S_p$) sometimes results in training samples producing low accuracy models. Moreover, omitting negative injection overfits the GNN to predict only positive edges. The findings of Abuoda et al. [2] show that eliminating single samples at random from the training set can have a positive impact on the accuracy. Therefore, I implemented a downsampling method that eliminates negative edge samples from the generated training set that GROWL would be trained on until $\Delta NP < x$, where x was tested to be either $25k$, $50k$, $60k$, or $65k$.

Alternatively, I followed the suggestion of Jiang et al. [71] and implemented ensemble learning, which is a technique similar to k-fold cross-validation. In ensemble learning, I generated the training set from randomly selected samples with *negative injection* as before, but instead of using the entirety of the training set, it is split into k folds. Next, I trained k models, and for each model, a fold is left out from the training set - which would be the validation set in case of cross-validation. During prediction, I calculate a label for the GNN's edges with all k models and calculate the agreement between the predictions by taking the arithmetic mean of the binary output labels and rounding the result. I have tested k values of 2, 3, 5, and 10.

I tested the presented two sample balancing techniques and their variations on the SALSA [5] dataset and its subsets, the RICA dataset, and the test set used by Thompson et al. [135] taken from the SALSA-CPP set (CPP-T). The training and testing were performed following the same protocol as described in Section 5.4.

The results of this evaluation can be seen in Figure 5.6. The best performing downsampling technique was the $x = 25k$ one which achieved an average F_1-score of $\overline{F_1} = 56.3\%$ and average standard deviation of $\overline{\sigma} = 41.9$ across all test sets. The other, higher x values performed poorly both in terms of F_1-scores and standard deviation.

Based on my evaluation, a 5-fold ensemble learning - although taking longer to train and predict with - outperforms both the tested downsampling methods and the GROWL algorithm not using sample balancing in terms of both average F_1-scores ($\overline{F_1}$), and standard deviation (σ). On average, this method was better by $\Delta\overline{F_1} = 7.38\%$ and $\Delta\sigma = -11.7$ compared to the original GROWL algorithm.

Based on the above-described extensive investigation, I selected a 5-fold ensemble learning approach to improve the performance of the GROWL algorithm. The new approach is termed Improved GROup detection With Link prediction (iGROWL). This approach minimises the chance of an "unlucky" training sample producing low-accuracy models and as a result, minimises the observed standard deviation.

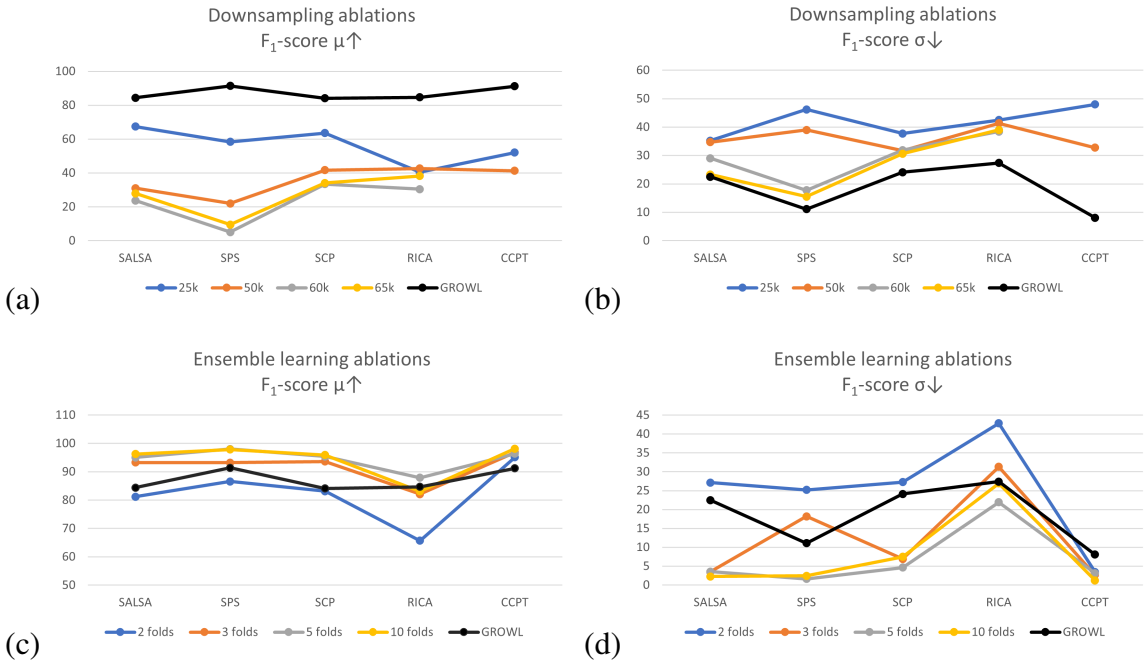


Fig. 5.6 These figures show the measured mean F_1-scores (μ) (a, c) and standard deviations (σ) (b, d) of the implemented sample balancing techniques. (a) and (b): Training GROWL on a training set downsampled to a number of samples satisfying the condition $\Delta NP < x$, where $\Delta NP = S_n - S_p$ represents the difference between positive (S_p) and negative (S_n) sample counts, and where x was tested to be either $25k$, $50k$, $60k$. (c) and (d): Ensemble learning with 2, 3, 5, and 10 folds as compared to original GROWL. The evaluation on CPPT does not include $x = 60k$ and $x = 65k$ as the maximum difference between positive and negative samples when training on the last 372 images of SALSA Cocktail Party (SALSA-CPP) is $51k$. SALSA: Synergetic sociAL Scene Analysis dataset [5]; RICA: RICA dataset; SPS: SALSA Poster Session (SALSA-PS) subset; SCP: SALSA Cocktail Party (SALSA-CPP subset); CPPT: First 64 samples of the SALSA Cocktail Party (SALSA-CPP) subset following Thompson et al. [135]'s evaluation practice.

Table 5.1 Comparison of GROWL and iGROWL against the state-of-the-art group detection methods, GCFF [124], REFORM [60], and Thompson et al. [135]'s model (T-GNN) in terms of mean F_1-scores ($\overline{F_1}$) on the SALSA [5] and RICA datasets. SALSA-PS: SALSA Poster Session subset; SALSA-CPP: SALSA Cocktail Party subset; CPP-T: First 64 samples of SALSA Cocktail Party following Thompson et al. [135]'s evaluation practice; σ: standard deviation of F_1-scores, calculated over 30 repeated evaluations.

Detection algorithm	SALSA		SALSA-PS		SALSA-CPP		RICA		CPP-T	
	$\overline{F_1}\uparrow$	$\sigma\downarrow$	$\overline{F_1}\uparrow$	$\sigma\downarrow$	$\overline{F_1}\uparrow$	$\sigma\downarrow$	$\overline{F_1}\uparrow$	$\sigma\downarrow$	$\overline{F_1}\uparrow$	$\sigma\downarrow$
GCFF	59.7%	9.5	64.7%	12.4	59.3%	8.4	52.1%	30.1	–	–
T-GNN*	–	–	–	–	–	–	–	–	70%	37
REFORM**	–	–	81.2%	–	–	–	–	–	–	–
iGROWL	95.1%	3.6	97.9%	1.6	95.4%	4.6	87.9%	21.9	96.4%	3.1
GROWL	84.4%	22.5	91.4%	11.1	84.1%	24.1	84.7%	27.4	91.2%	8.1

*Taken from [135]

**Taken from [60]

5.4.4 Experimental results

Table 5.1 shows the average F_1-scores for predicted groups across test sets taken from the SALSA dataset [5], its subsets Poster Session (SALSA-PS) and Cocktail Party (SALSA-CPP) as well as the RICA dataset. Note that GROWL and GROWL with a 5-fold ensemble learning (iGROWL) were both trained on a random, 60% split of the SALSA-PS set only. To showcase that the resulting F_1-scores produced by GROWL were not a result of a lucky training set selection, I trained and evaluated both GROWL and iGROWL 30 times, randomly shuffling and selecting new training sets from SALSA-PS on each occasion. The results of this evaluation can be seen in Table 5.1. Looking at the results, the generality of GROWL across different test sets is similar to that of GCFF [124]; however, it performs significantly better, with an average of 20.8% across all evaluated test sets. Moreover, GROWL outperforms the previous state-of-the-art method, REFORM [60] when evaluated on SALSA-PS by a margin of 10.2% in terms of F_1-scores. Lastly, GROWL showed significant improvement of $\Delta\overline{F_1} = 21.2\%$ and $\Delta\sigma = -33.9$ compared to the reported F_1-scores and standard deviation of T-GNN respectively. I hypothesise that T-GNN is performing worse due to the missing representation learning component, which in GROWL and iGROWL generates feature embeddings based on the structure of the entire scene. Without such representation, the node features alone do not carry global information for the pairwise predictor to reliably identify groups.

Examples of predicted interaction groups from the SALSA-CPP and RICA datasets can be seen in Figure 5.7. I observed that GROWL is good at detecting separate, unconnected

Fig. 5.7 These figures show the RGB image (left), ground truth (middle) and GROWL predicted (right) graph representations of a frame from the SALSA Cocktail Party dataset (a-b) and the RICA dataset (c-d).

groups. However, on some occasions, it connects two groups through a single node, producing both false positive and false negative group detections with a single mistake.

As described in Section 5.4.1, I also evaluated GROWL with automatically detected bounding boxes on the RICA dataset (see Table 5.2). Given the assumption that YOLOv4 [17] could detect all individuals in a scene, GROWL achieved an average F_1-score of 74.7. Without the above-mentioned assumption, GROWL's average was 2%. These findings indicate that the reliable detection of individuals has a strong impact on the group detection performance, which was also observed when investigating the results of the Agglomerative Hierarchical Clustering based unsupervised method presented in Section 4.4. While GROWL can identify interaction groups accurately, the inaccuracies of bounding box detection influence its performance significantly. In other words, GROWL's group detection

Table 5.2 Comparison of GROWL against GROWL without orientation features (GROWL-O), in terms of mean F_1-scores ($\overline{F_1}$) on the SALSA [5] and RICA datasets. SALSA-PS: SALSA Poster Session subset; SALSA-CPP: SALSA Cocktail Party subset; RICA-Y: GROWL tested on the RICA dataset with bounding boxes automatically detected using YOLOv4 [17]; σ: standard deviation of F_1-scores, calculated over 30 repeated evaluations.

Detection algorithm	**SALSA** $\overline{F_1}\uparrow$	**SALSA** $\sigma\downarrow$	**SALSA -PS** $\overline{F_1}\uparrow$	**SALSA -PS** $\sigma\downarrow$	**SALSA -CPP** $\overline{F_1}\uparrow$	**SALSA -CPP** $\sigma\downarrow$	**RICA** $\overline{F_1}\uparrow$	**RICA** $\sigma\downarrow$	**RICA-Y** $\overline{F_1}\uparrow$	**RICA-Y** $\sigma\downarrow$
GROWL-O	22.5%	39.6	33.7%	16.5	30.9%	41.5	44.5%	43.2	52.2%	43.7
GROWL	84.4%	22.5	91.4%	11.1	84.1%	24.1	84.7%	27.4	74.7%	41.9

accuracy is significantly hindered if YOLOv4 [17] fails to detect people in the scene since undetected individuals introduce False Negative node classifications when calculating F_1-scores.

Based on the presented repeated evaluation, it can be observed that the standard deviation (σ) of measured average F_1-scores ($\overline{F_1}$) was high. This is due to the different randomly selected training sets and the resulting varying amount of positive and negative edges GROWL is trained on. According to the performed evaluation, this effect can be so radical that up to 2 of the 30 measurements will result in low ($\sim 11\%$) mean F_1-scores when testing on SALSA, SALSA-CPP and RICA. These findings indicate that the single training and evaluation that Hedayati et al. [60] reported is not sufficiently reliable, as the selection of training set has a significant impact on the performance of the group detection accuracy.

To investigate cases when the GROWL model produces low accuracies, I analysed the sample distribution it was trained on with different training samples. I observed that the number of positive edge samples ranges between 23200 and 23850 while the number of negative edge samples is between 88000 and 90250. Throughout different training sets the ratio between positive and negative samples is on average 20.8% and 79.2%, respectively. However, when there are more than 90000 negative samples, the measured mean accuracy drops significantly. To address this issue, I implemented several sample balancing techniques (see Section 5.2.4), and concluded that 5-fold ensemble learning (iGROWL), while requiring more computation time, helps in improving the mean accuracy and diminishes the standard deviation across repeated tests. As a result, iGROWL produced the highest measured mean F_1-scores with an average improvement of $\Delta\overline{F_1} = 7.38\%$ and $\Delta\sigma = -11.7$ scores across all evaluated datasets compared to the reported values of GROWL.

5.5 Ablations on GROWL

In this Section, the importance of the orientation feature is investigated in connection with detecting interaction groups. It also presents the effect of calculating the orientation of people based on the recorded depth modality of a robocentric view. Lastly, it investigates the contribution of negative injection.

5.5.1 Contribution of orientation features

While the SALSA dataset [5] provides the ground truth for orientation information, for the RICA dataset this information is not available. Therefore, as explained in Section 5.3.1, I used the Deep-orientation method [85] for automatically estimating individuals' orientation from depth images. To demonstrate the reliability of orientation estimation and its contribution to accuracy, GROWL was trained and tested through 30 iterations as described in Section 5.4.4, using position information only for calculating the node embeddings, excluding the orientation. The resulting mean F_1-scores are provided in Table 5.2.

I observed that the group detection accuracy dropped significantly without the orientation information. Based on these findings, I hypothesise that GROWL creates an encoded representation similar to what Vascon et al. [140] described as attention frustum to identify interaction groups. Without the orientation information, all evaluation accuracies deteriorated significantly. However, this change was less significant in both RICA and RICA-Y. This could be the result of noise introduced to raw node features during the graph representation calculation which translates the robocentric view to a third-person view. This indicates that while the orientation information from a robocentric view carries added noise, it still contributes significantly to the overall performance of the model when included as a feature.

5.5.2 Contribution of negative injection

Following the findings of Zhang and Chen [155], I employed negative injection in order to achieve a model which can generalise better. In this section, I verify that in the context of interaction group detection the exclusion of negative sampling results in overfitting, making the model prone to only predicting existing edges.

To investigate the contribution of training and evaluating with negative injection, as before, I trained and evaluated GROWL 30 times as described in Section 5.4.4 without generating training graphs via negative injection – only taking graphs' existing (positive) edges. As expected, all evaluations concluded that the trained models achieve average F_1-scores of near 0%. Upon further investigation, I confirmed that this is due to the model producing fully-connected graphs, which – with the tolerance of $T = 2/3$ used during

evaluation (see Section 5.4.2) – results in scores of 0 regarding both precision and recall scores.

5.6 Discussion

While at the time of writing Improved GROup detection With Link prediction (iGROWL) is the state-of-the-art group detection method, there are still improvements to be made to account for some of its faulty link predictions. As illustrated in Figure 5.7, both GROWL and iGROWL can inadvertently connect individuals who in reality don't belong to the same group. To address this, self-supervision could be introduced to determine strong and weak edges.

While this work demonstrated the benefits of using GNNs for supervised group detection, it did not investigate their robustness to errors. As highlighted in Chapter 4, an AHC-based method's performance is hindered by inaccurate detections. While the presented results (see Section 5.4.4) show that iGROWL performs well with bounding boxes detected with YOLOv4 [17], its sensitivity to inaccurate position or orientation information has not been investigated. Therefore, while iGROWL's robustness to non-GT inputs needs to be investigated, a promising path would be to extend iGROWL into an end-to-end approach that can take an image of a scene as input and jointly perform person and group detection.

Finally, iGROWL could be jointly used with Graph Neural Network based node [121] and subgraph classification [8] techniques to also identify F-formation types after computing interaction groups.

5.7 Conclusion

I proposed a Graph Neural Network (GNN) based approach called GROup detection With Link prediction (GROWL), to learn the links between individuals and their association with groups. GROWL learns the structural relationships between individuals through graphs using individuals' position and orientation information to generate feature embeddings via GraphSAGE [57] and use them as node features. Pairs of these node features are given as inputs to a shallow Multi-Layer Perceptron (MLP) for predicting the existence of edges between the inspected nodes.

GROWL outperforms previous state-of-the-art approaches such asGCFF [124], RE-FORM [60], and Thompson et al. [135]'s GNN based solution while maintaining the same generality.

The ablation studies presented in Section 5.5 verified that orientation information is an important feature for interaction group detection. I also demonstrated that in the case of robocentric images, calculating orientation from RGB and depth data is possible without significantly affecting group detection accuracy. Lastly, I verified that negative injection is crucial for training the model without overfitting, in line with the findings of Zhang and Chen [155].

Throughout the evaluation of GROWL, I measured large standard deviations and observed that the selected training set has a significant impact on group detection accuracy. Firstly, this showcased the importance of evaluating this method based on repeated evaluation measurements. Secondly, it lead to the investigation of sample imbalance and the exploration of viable sample balancing techniques. As a result, the proposed algorithm was improved by applying a 5-fold ensemble learning method. This improved version of GROWL (iGROWL) improves both the performance and consistency of the original algorithm.

Chapter 6

Socially-Aware Robot Navigation

Automatic navigation in both static and dynamic environments has been solved with reliable accuracy by the current state-of-the-art, research focus has only recently shifted to socially-aware navigation. Socially-aware navigation, by the definition established by Rios-Martinez et al. [119], is a navigation strategy executed by a robot that follows social conventions. In this case following social conventions primarily entails good space management in order to make the interaction with humans comfortable by being adaptable and predictable.

The following Sections explore the most recent state-of-the-art algorithm (see Section 6.1) and present my proposed solution for advancing the field to achieve a more socially-aware solution (see Section 6.3) as well as to provide a thorough, more unified evaluation method for future comparability (see Section 6.4).

6.1 Analysis of the State of the Art

As detailed in Section 2.3, while there have been a number of approaches addressing socially-aware navigation, only a few of them incorporate group information in their models as they often consider groups as impassable dynamic objects [36, 74, 89]. This results in a strong abstraction of the scene. Furthermore, the most recent state-of-the-art approaches have started using Reinforcement Learning (RL) techniques [25, 36, 74, 141], primarily actor-critic learning-based approaches [36, 74], as opposed to rule-based fuzzy logic controllers [40, 83, 100, 107] and social force and zone based models [106], achieving the highest scores in social navigation challenges. The most recent approaches [36, 66, 73, 89] started incorporating group information into their models to improve social competence, as disrupting interaction groups by crossing through them may be considered anti-social [73].

The works of both Katyal et al. [74] and Do et al. [36] were both based on RL algorithms, but according to the evaluation presented by Katyal et al. [74], they did not

train or test on a dataset labelled on a group-level. Instead, their work used a Poisson distribution [27] to randomly assign pedestrians close to each other to groups in a scene. Therefore, in this work I consider the solution presented by Do et al. [36] as the latest state-of-the-art within the field of Reinforcement Learning based socially-aware navigation, which attempted to improve social competence by incorporating group information.

Do et al. [36] proposed a framework based on actor-critic learning. Their aim was to navigate to a goal in an environment through a crowd. They presented a framework that takes into account both the personal space of individuals [56], F-formations of groups [75], and human-object interaction space [136]. This solution enables a robot to receive information in a 240° angle in front of itself from three sources. The authors monitored the robot's distance from obstacles, human and social interactions, and the goal. Lastly, they tracked the robot's position. Using these inputs, they trained an Asynchronous Advantage Actor-Critic (A3C) network to predict output commands which can be used to control the robot's linear (v) and angular (w) velocities. They capped these velocities to fall between $v \in [0.0, 1.2]$ and $w \in [\frac{-\pi}{6}, [\frac{\pi}{6}]$, not allowing the robot to move backwards due to the lack of range coverage in that direction. They proposed a reward function [36]:

$$r_t^r = r_t^{ro} + r_t^{rh} + r_t^{rs} + r_t^{rg} \tag{6.1}$$

where r_t^r is the overall reward at a given timestep. r_t^{ro} is a penalty for colliding with obstacles, r_t^{rh} is a penalty for colliding with individuals or approaching people too close to comfort based on the idea personal space (0.9 m) defined by Hall [56], r_t^{rs} is a penalty for crossing interaction spaces, and r_t^{rg} is a reward for getting closer to the goal compared to the previous timestep, or reaching it.

They defined social interaction spaces by taking the centre of a group, positioning the individuals in the interaction on the perimeter of a circle (o-space), which corresponds to the p-space according to the definition of Kendon [75] as seen in Figure 2.1(b), and calculating the radius of the circle. The robot was penalised when its distance from the centre of the group was less than the group radius. The detailed description of the reward functions can be seen in the paper of Do et al. [36], and their parameters used for calculating rewards are presented in Table 6.1.

The researchers trained and evaluated their solution on a generated simulated environment created in the Stage simulation suite [50], controlling simulated human behaviour with a social force model [62]. They used the Social Individual Index (SII) proposed by Truong et al. [137], which measures the robot's distance from individuals and compares it to a physical and a psychological distance threshold defined by Hall [56].

The implementation and evaluation described above make significant abstractions regarding how groups as social entities are processed in a scene. According to their

Table 6.1 Parameters for training and testing the actor-critic algorithm [36]. α: learning rate; γ: discount factor; t_{update}: frequency update; t_{max}: maximum steps in an episode; N_s: input size; N_a: action size; r_0^{ps}: human personal space; $r^o_{collision}$: punishment for colliding with obstacles; $r^h_{collision}$: punishment for colliding with people; $r^{ps}_{collision}$: punishment for entering personal space; $r^s_{collision}$: punishment for entering social zones; r_{tgoal}: reward for getting closer to the goal; $r^h_{arrival}$: reward for reaching the goal

Parameter	Value	Parameter	Value	Parameter	Value
α	0.001	γ	0.99	t_{update}	20
N_s	722	N_a	14	r_0^{ps}	$0.9(m)$
t_{max}	500	$r^o_{collision}$	-5	$r^h_{collision}$	-10
$r^{ps}_{collision}$	-0.2	$r^s_{collision}$	-0.2	$r^h_{arrival}$	10
r_{tgoal}	0.1				

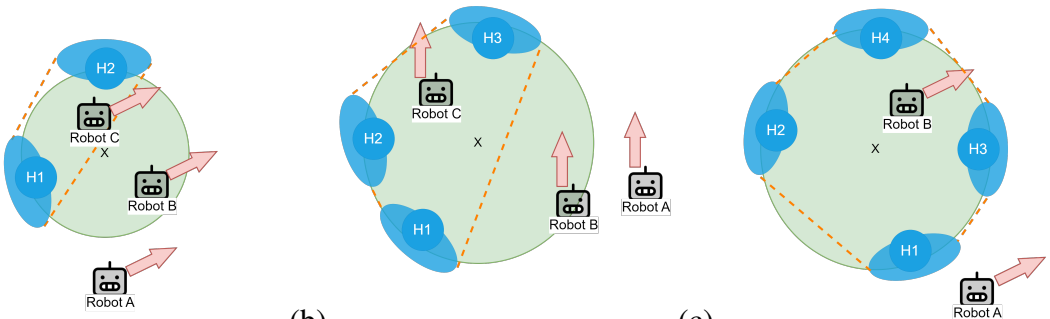
social reward function targeting groups, a robot approaching an *L-arrangement* group formation [94] (see Figure 2.1(a-1)) would be considered to breach it if it was on the opposite side of the group centroid than the individuals but closer than the group radius. This is inaccurate as, especially in densely crowded scenarios, it constantly penalises the robot approaching groups, even if it does not violate individuals' physical or psychological comfort zones. Examples of the robot not disturbing and disturbing group interaction in comparison to the circular group representation proposed by Do et al. [36] can be seen in Figure 6.1.

Moreover, the proposed state-of-the-art approach of Do et al. [36] was only evaluated on a generated environment, using only the two thresholds described by SII [137]. According to my review of the evaluation of socially-aware navigation (see Section 2.3.3), this metric is not reliable enough and represents a limited estimate of the social competence of a robot in light of all other investigated quantitative and qualitative metrics.

6.2 Simulation environment

As described in Section 2.3.3, most previous approaches either use computer-generated simulations for evaluating their solutions or test in real-world environments. However, there is a gap between purely simulated environments and real settings due to inevitable abstractions [92, 77]. Therefore, I investigated two recent, real-world datasets used for group interaction scenarios, which can be used for creating real-to-sim simulations and more realistic training and testing scenarios.

The Synergetic sociAL Scene Analysis (SALSA) dataset [5] can be used to create simulated environments based on the position and orientation of people, however, these



(a) (b) (c)

Fig. 6.1 These figures showcase that sometimes, despite entering an area defined by the group centroid (marked with 'X' in green circles) and the group radius, the robot does not necessarily enter the conversation space defined by the orange dashed lines. In (a) and (b) Robot A is not breaching the conversational space according to the penalty for crossing interaction spaces (r_t^{rs}) proposed by Do et al. [36]; Robot B does not breach the conversational space but is penalised; and Robot C breaches the conversational space. This relation between Robot A and B can be distinguished in smaller groups like L-shaped (a), Triangular (b) formations. However, it's not feasible (c) in larger, Circular groups. In (c), Robot A is not penalised and is outside of the group space, and Robot B cannot be in a place where it is in the group's circle but outside of the conversation space.

settings will not be responsive to the movements of robots traversing them. There is currently no available software that can be used in its default state to create a simulation based on real group data, which is 3D, represents people as avatars, and also provides agent behaviour for simulated individuals. However, Grzeskowiak et al. [52] proposed a simulation-based benchmark tool called CrowdBot, which enables human behaviour simulation in generated crowds. While this tool does not support the setup of real-to-sim environments, it fulfils all other aforementioned criteria. They utilise either a social force-based navigation algorithm titled Pedsim [62] or the optimal reciprocal collision avoidance (ORCA) [139] model to simulate agents' behaviour. Grzeskowiak et al. [52] argue that the CrowdBot simulation can be a tool for training and evaluating robot navigation approaches as in many regards it is representative of a real crowd's behaviour.

Based on the basic CrowdBot environment (see Figure 6.2[1]), I have changed the robot from the default automatic wheelchair model the model of the Pepper robot [110] and disabled simulated camera sensors in the simulation. The Pepper robot was chosen as it is an embodied mobile robot that is widely used for Human-Robot Interaction (HRI) scenarios. The CrowdBot simulation [52] does not have other similarly capable, HRI-ready robots implemented. The simulated cameras of the Pepper robot were disabled because the state-of-the-art algorithm and consequently my proposed method do not rely on camera data. In a real-world setting, the usage of camera data will be necessary to produce the

[1]Taken from the Technical Report of Safe Robot Navigation in Dense Crowds [115]

Fig. 6.2 The default CrowdBot simulation setup, where a robotic wheelchair needs to navigate from left to right until the end of the corridor.

required feature vectors, but in simulation, the required information can be computed, as the positions of scene elements and agents are always known.

The outline and functionality of the simulation's different modules can be seen in Figure 6.3. Leveraging the capabilities of Unity3D [55], I implemented the 240° distance readings described by Do et al. [36] (see Section 6.1) via *Raycast*s and used the *Tag* system of the development environment to identify walls as obstacles, people, and the goal to read their distance in relation to the robot and structure these readings in a 3×240 dimension feature array. As for the robot's position features, I used the model's *Transform* position provided by Unity3D.

The CrowdBot environment communicates with ROS [116] via a *websocket*. This allows Unity3D to publish messages to a *python* backend (ROS backend) which processes the received messages, based on the timestamp calculates individuals' group IDs, and calculates rewards for the Reinforcement Learning (RL) algorithm which is trained and used for inference by the algorithm training and testing module (RL backend).

The RL backend is disjoint from the ROS backend, as the actor-critic implementation requires PyTorch [111], which runs on *python3*, but the CrowdBot implementation is written to interface with *python2*, making the two incompatible. ROS requires time to progress in order to publish and receive messages, enabling communication between the modules of Unity3D, ROS backend, and RL backend by using a so-called "clock". As the *clock* needs to represent time passing in real-time, but actor-critic algorithms need to "step" time, a simulated clock (*sim-clock*) was introduced.

This *sim-clock* is in charge of incrementing time in the simulation, allowing all members of a scene to move. Simulated time is stepped forward when the RL backend computes an action to submit to the robot through the ROS backend. As a result, Unity3D applies the submitted velocity command to the robot, generates behaviour for the human agents, and computes a feature vector which is returned to the ROS backend. The ROS backend relays the feature vector and computes rewards to the RL backend. These features and the reward are then used by the RL backend to compute an action for the robot in the next timestamp, completing the cycle for a single time increment.

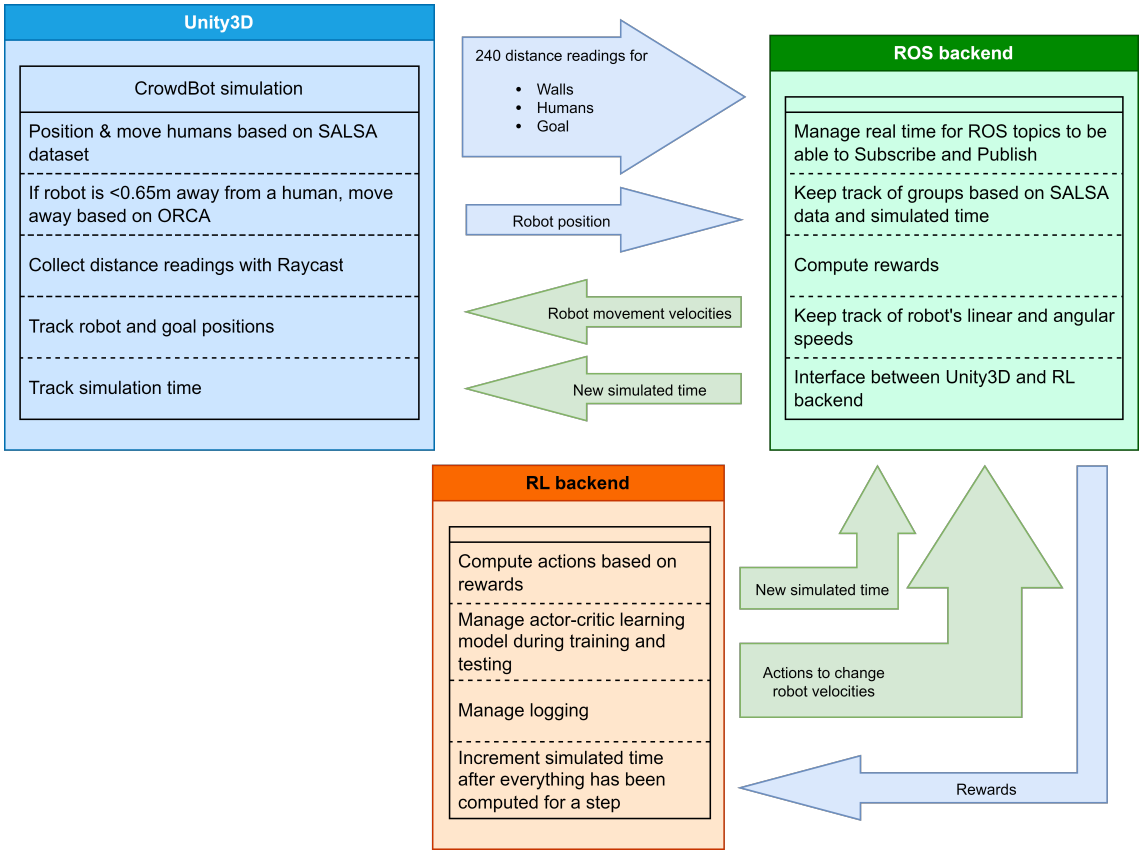


Fig. 6.3 This figure shows the three modules which make the CrowdBot simulation environment work with an actor-critic algorithm. The modules detail their main functionalities and the arrows indicate information being passed between them via ROS topics.

In Unity3D, human positions are regulated based on the position and orientation information of the SALSA subsets, SALSA-PS and SALSA-CPP. The human agents can, however, divert from these recordings if the robot gets too close, in which case the agents navigate based on the ORCA [139] avoidance implemented in the CrowdBot simulation. I defined "too close" as 0.65 m, a value lower than the personal space used by Do et al. [36] in their reward (0.9 m), but still above the lower bound of the 0.45 m–1.2 m personal space defined by Hall [56]. This should allow the algorithm to learn avoidance so it doesn't necessarily disturb human participants of a scene, also allowing them to avoid the robot at a reasonable distance; i.e., before it enters the $<$0.45 m intimate space [56] of individuals. Despite these settings, the robot can still risk collision with people if it does not regulate its speed, as the simulated agents will not have time to react.

The proposed simulation environment serves as a suitable platform for both training and evaluating socially-aware navigation solutions and presents a simulated setting based on a real-world crowd dataset. While it may not be as dynamic as the original CrowdBot simulation [139], in this setting human agents are still positioned based on waypoints until

the robot's proximity triggers avoidance behaviour, which is achieved via the ORCA [139] algorithm. In the same fashion, CrowdBot's implementation also facilitates general crowd movement based on preset positions and orientations and handled disruptions via avoidance algorithms. Despite the fact that agents gesture more in the real-to-sim setting, as they remain stationary most of the time, the robot's distance is always calculated from their torso, just as in CrowdBot. The most significant difference between the original simulation and the one described above is the implementation of the RL and ROS backends. These enable the robot to compute features from sensor readings and train both the proposed and SOTA algorithms.

6.3 Advanced Actor-Critic based navigation

In this section, I introduce Socially-Aware Navigation between Groups (SANG) which is an Advantage Actor-Critic (A2C) learning based algorithm that takes into account the robot's static environment as well as human members of the scene. This approach is based on the same actor-critic network structure and reward values as the proposed state-of-the-art (SOTA) introduced by Do et al. [36], who used the same features in their Asynchronous Advantage Actor-Critic (A3C) method.

Based on the work of Mnih et al. [102], an actor-critic algorithm features two networks, the actor and the critic, which are responsible for choosing actions and evaluating the chosen action, respectively. An advantage actor-critic algorithm also includes an advantage function that calculates an agent's Temporal Difference (TD) Error. In principle, TD Error can be used for measuring if a chosen action is better or worse than expected, and if it is better, encourage an agent to make more of it. The asynchronous element of these networks is that multiple instances of them can be trained at once with parallel workers, updating the main network's weights with their aggregated change, thus allowing the RL algorithm to train quicker. However, with an ample number of epochs to allow convergence, an A2C algorithm can learn the same functionality as an A3C based one, just slower.

In the case of SANG, only A2C could be implemented due to the limitation of the CrowdBot simulation environment [52], namely that it can only run one instance of the simulation at a time. To create a comparable benchmark, I also retrained the algorithm of Do et al. [36] without the asynchronicity in the real-to-sim environment running in the CrowdBot simulation.

In comparison to the work of Do et al. [36], my solution treats group information at an individual level, not only taking into account blobs or circular areas represented by groups and group centres, but introducing a group label input instead for each individual in the scene. This representation is useful as it does not force the robot to go around groups,

as it is sometimes necessary to cross them in narrow, crowded environments. However, it can still be used to penalise the robot for doing so. Moreover, this representation is more realistic, as it does not assume that people are positioned on the perimeter of a circle, treating the entire circle area as if it belonged to the group, but groups are treated as convex polygons. The use of convex polygons instead of circular representations was also supported by the findings of Katyal et al. [73].

My proposed solution, following the practices of previous work in Reinforcement Learning based navigation [141, 25, 36, 74], was set up to be trained in a simulated environment to eliminate expensive real-world demonstration collection. I utilised CrowdBot [52], but instead of relying on its built-in highly dynamic scenarios or other generated simulation settings, I aimed to minimise the real-to-sim gap by mapping the Cocktail Party (SALSA-CPP) and Poster Session (SALSA-PS) subsets of SALSA [5] to the environment. In my setup, the simulated human positions are set by the SALSA position on every timestamp. However, if these agents get too close to each other, or the robot approaches them too closely, they perform avoidance behaviour based on CrowdBot's ORCA [139] implementation which comes built into the simulation environment.

In line with the work of Do et al. [36], I initially generated inputs for the A2C model based on three aspects of the scene. They proposed tracking 240 rays in front of the robot, and in each of those measuring the distance to walls, humans, and goals, creating a 3×240 array of floating point values. Moreover, they used two more inputs, representing the robot's x and y coordinates in the scene. In addition, I introduced the tracking of an individual's group membership in these 240 directions, meaning that if a person is detected by a Raycast, I do not only use their distance to the robot, but also provide their group ID as an input feature. This results in a 1 dimension bigger feature array, resulting in 4×240 features plus the robot's x and y coordinates.

Based on the above-mentioned feature vector, I proposed an Advantage Actor-Critic (A2C) model for training a socially-aware navigation policy. The structure of the A2C model is in line with the work of Do et al. [36] and can be seen in Figure 6.4. I highlight that in contrast to the SOTA, I chose the range of linear (v) and angular (w) velocities to be $v \in [-0.5, 1.25]$ and $w \in [\frac{-\pi}{3}, [\frac{\pi}{3}]$, as opposed to the ones presented by Do et al. [36] (see Section 6.1), to match the distances and human speeds of the CrowdBot environment better. Moreover, I allowed the robot to have negative linear velocity, as reversing in a densely crowded space can be beneficial to avoid collisions with people.

I defined the reward function on every timestep t based on Do et al. [36]'s reward presented in Section 6.1, Eq. 6.1. While keeping the reward scoring in line with the SOTA algorithm [36]'s, due to the inclusion of individual-level group labels, I introduced a new social interaction reward, r_t^{rs}. Previously, this reward was calculated based on whether the

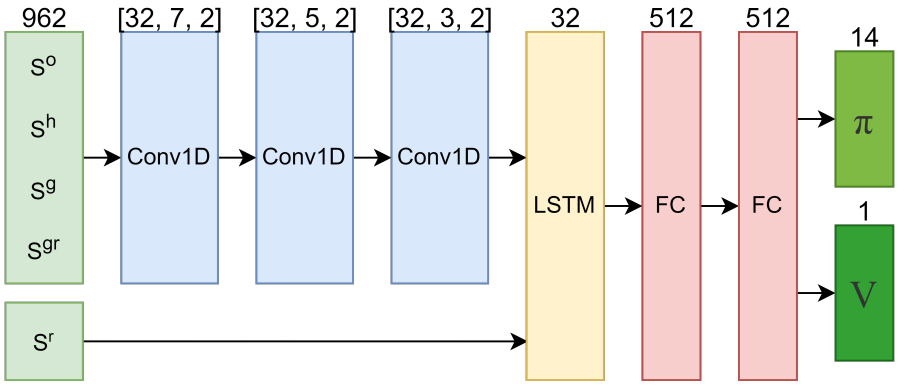


Fig. 6.4 Representation of the policy (π) and value (V) networks. S^o, S^h, and S^g are the distance measurements for obstacles, people and group centres respectively. S^{gr} represents the group label of people around the robot, and S^r is the position of the robot. The convolutional (Conv1D) layers are characterised by the indicated [filters, kernel size, stride], the LSTM layer has 32 hidden states, the fully connected (FC) layers have 512 neurons each. The output of the policy network (π) is the $N_a = 14$ action space of the robot, and V represents the single value output of the value network.

robot was within the radius of a group, measured from the group centroid, thus treating groups as circular entities. Instead, I propose the reward term as:

$$r_t^{rs} = \begin{cases} r^s_{collision}, & \text{if } d_t^g \leq d_t^h \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

where d_t^g is the sum of the robot's Euclidean distance (see Eq. 5.1) from all group members, and d_t^h is the sum of Euclidean distances between pairs of group members. This relationship allows the robot to go close to the convex group boundaries, and it permits the robot to enter o-spaces of groups, but it penalises the latter (see Figure 6.5). This way of computing group boundaries is more accurate, as the robot only breaches a boundary if it crosses the imaginary line between two individuals who are in a group, not when it enters a circle drawn around all individuals. The conditions of Eq. 6.2 were chosen to be noise free and without a margin of error as the SOTA implementation also defined their reward term in this fashion. This is supported by the real-to-sim implementation of the training environment, as the distance readings, and consequently, the input features can also be noise-free.

SANG was trained with the same parameters as outlined by Do et al. [36] (see Table 6.1) on the first half of the Cocktail Party set of the SALSA dataset [5] until its reward score converged. The only training parameter changed was the goal-reaching reward ($r^h_{arrival}$), which was set from 10 to 100 in order to make the robot more consistent at reaching its goal.

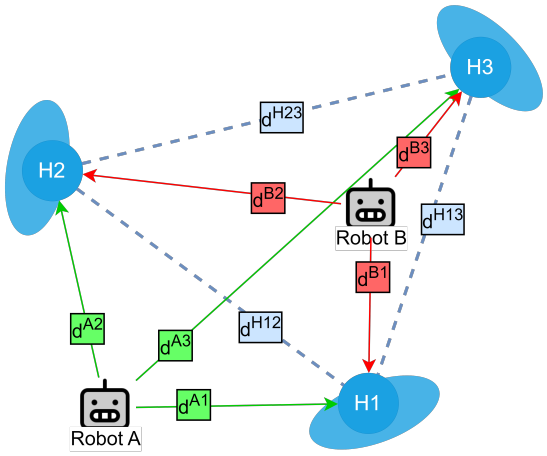


Fig. 6.5 A representation of how the social interaction reward (r^{rs}) is calculated in two scenarios (Robot A and Robot B) given an interaction group formed by 3 people (H1, H2, H3). Blue dashed lines indicate the interaction space of the group. As described in Equation 6.2, distance between group members can be calculated by $d_t^h = d^{H12} + d^{H13} + d^{H23}$, and the robots' distance by $r_t^g = d^{A1} + d^{A2} + d^{A3}$ and $r_t^g = d^{B1} + d^{B2} + d^{B3}$ for robots A and B, respectively. Based on the relation between r_t^g and d_t^h, Robot A will not receive a penalty ($r_t^{rs} = 0$) as $r_t^g > d_t^h$, while Robot B will receive a penalty $r_t^{rs} = r_{collision}^s$ as $r_t^g < d_t^h$.

6.4 Evaluation procedure

To evaluate the solution, I used two parts of the SALSA dataset [5]. The data of the Poster Session (SALSA-PS), and the second half of the Cocktail Party (SALSA-CPP) the models were not trained on. It is important to note, that for all training and test procedures GT data was used to determine and set the position and orientation of human agents in the environment. This is considered a common procedure in the related literature. Moreover, since in SALSA-CPP people are standing closer to each other, I reduced the width of the navigation area, making this test set more challenging. By having a smaller area in the setting, the robot could not navigate around the entire crowd anymore, forcing it to perform avoidance in between the groups. In all test settings, the aim was to reach a goal in a randomly generated position. The range of possible locations was chosen in a way to ensure that it's necessary to approach groups and so that in many cases it's required to cross the entire length of the crowd. Screenshots of the two environments can be seen in Figure 6.6.

To perform a thorough evaluation, and since the SOTA algorithm of Do et al. [36] did not use most of the metrics proposed by Mirsky et al. [101] or Gao and Huang [47], I compared data collected from three sources in the two settings presented above. I asked (1) human participants to navigate the crowd in the simulation (HUMAN); I ran the (2) non-asynchronous, Advantage Actor-Critic (A2C) SOTA algorithm of Do et al. [36] trained

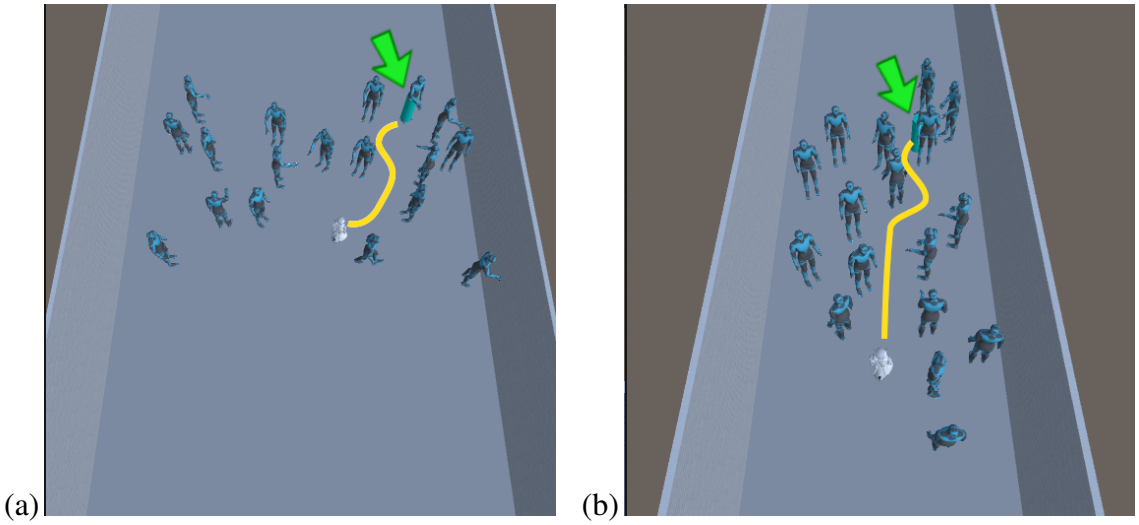


Fig. 6.6 This figure shows the settings of the (a) SALSA Poster Session (SALSA-PS) and (b) SALSA Cocktail Party (SALSA-CPP) test scenarios. The white agent is the robot, the blue agents are simulated humans positioned based on the SALSA dataset [5], and the teal cylinder (highlighted by the arrow pointing at it) is the randomly generated goal position. The yellow line indicates a path the robot might follow to perform socially-aware navigation.

on the first half of the SALSA-CPP setting; and I ran (3) my proposed, Advantage Actor-Critic (A2C) based algorithm with the improved group information representation (SANG) trained on the first half of the SALSA-CPP setting as well.

My HUMAN tests involved 10 participants controlling the robot via a keyboard. The data collection procedure can be seen in Figure 6.7. Participants were tasked with navigating to reach a randomly generated goal, e.g., navigating to the teal cylinder in Figures 6.6(a) and (b), producing the indicated yellow path. They were required to perform navigation in both the SALSA-PS and SALSA-CPP test settings, three times in each. A trial run was finished by either reaching the goal or crashing into a wall or a simulated human. Participants had the opportunity to familiarise themselves with the task, the control, and the behaviour of the controlled robot for 15 minutes. Furthermore, before data collection, participants were required to perform at least 5 successful navigation trials on the training set of the SALSA-CPP setting.

I evaluated the SOTA and SANG algorithms 100 times in both test settings. Just as in the case of the HUMAN tests, a trial run could finish by the algorithm either reaching a goal or colliding with the walls or simulated humans of the setting.

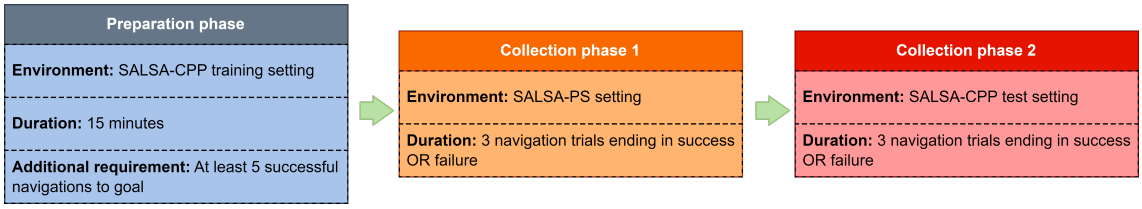


Fig. 6.7 This figure details the three phases participants were required to do for the HUMAN tests to establish human-generated socially-aware navigation benchmarks in the SALSA Poster Session (SALSA-PS) and Cocktail Party (SALSA-CPP) settings.

6.4.1 Evaluation metrics

Following the work of Mirsky et al. [101] and Gao and Huang [47], the presented evaluation recorded multiple, commonly used quantitative metrics in all three tested navigation outputs. The selected metrics can be seen in the list below.

- Successful navigation – reaching the goal without colliding with walls or humans;
- Smoothness – the *distance travelled* divided by the *angle turned*
- Minimum distance to humans – how close did the robot approach humans;
- Minimum distance to group centroids – how close did the robot approach the centres of interaction groups;
- Group O-space breach count – how many time-steps the robot spent within the bounds of an O-space [75] of a group.

In addition to the quantitative evaluation, for a more thorough overview of social compliance comparison, I also measured the social awareness of the behaviour learned by the algorithms via two qualitative approaches following the work of Gao and Huang [47]. I created videos of all three ways of navigation in both real-to-sim test settings. I used the HUMAN navigation videos paired with either of the algorithm outputs to perform a Navigation Turing Test proposed by Devlin et al. [35]. Within this test, participants were asked to look at pairs of videos taken in the same setting, and decide which one looks more human-like while navigating, a HUMAN navigation or one generated by the SOTA or SANG algorithms. Moreover, they were asked how confident they feel in their choice or if they had further comments in case participants wished to elaborate on the reasons behind their answers. To my knowledge, this has been the first time the Navigation Turing Test was applied in a real-to-sim crowd navigation evaluation since its presentation by Devlin et al. [35]. The data collection protocol was approved by the Ethical Committee of King's College London, United Kingdom (Review reference: MRSP-21/22-32163).

Table 6.2 Quantitative Evaluation Results

Navigation method	HUMAN		SOTA		SANG	
SALSA test setting	CPP	PS	CPP	PS	CPP	PS
Success rate ↑	93.33%	96.67%	46.0%	36.0%	35.0%$^{\bullet}$	39.0%
Mean Smoothness ↓	2.10	3.85	8.38	5.40	0.82°	0.72
Mean minimum distance to humans↑	0.61	0.66	0.19	0.24	0.58$^{\circ\star}$	0.56°
Mean minimum distance to group centroids ↑	1.06	0.26	0.88	0.08	0.93°	0.13°
Mean Group O-space breach count	3.93	53.97	28.16	16.18	40.60$^{\bullet}$	31.49°

$\circ$ or $\bullet$: statistically significant ($p < .05$) improvement or degradation when comparing SANG with SOTA given HUMAN benchmarks;
$\star$: no significant difference ($p > .05$) compared to HUMAN benchmark.

I also selected one video from both SALSA-PS and SALSA-CPP settings, generated by running the SOTA and SANG algorithms and had participants answer a Perceived Social Intelligence (PSI) questionnaire [11] based on them. The videos were selected depicting successful navigation samples to a similarly placed goal position for better comparison of the algorithms. Moreover, only one video was used per setting and per algorithm in order to minimise noise from observing navigation to different goal positions. Based on these results, I measured perceived social awareness, compliance, and friendliness.

6.4.2 Experimental results

Based on the quantitative measurements, I acquired the metrics describing how humans navigated the two test settings and compared them to the performance of the state-of-the-art (SOTA) and my proposed Socially-Aware Navigation between Groups (SANG) algorithm. The results from the evaluation for all metrics outlined above in Section 6.4.1 can be found in Table 6.2.

It can be observed that while the SOTA algorithm performed better, neither algorithm could compare to human accuracy when it came to reaching the goal position. I measured significant differences between the recorded metrics of exhibited behaviours by performing an *comparison of means* test (independent samples t-test [39]), which calculates the difference between the observed means in two independent samples taking into account the standard deviation. The only metric not exhibiting significant difference was the *Minimum distance to humans*, where SANG was not significantly different ($p < 0.05$) from the human-generated values with p-values of 0.85 and 0.26 for the SALSA-CPP and SALSA-PS settings, respectively. In this regard, the SOTA algorithm performed significantly worse compared to the others.

(a)

Navigation Turing Test HUMAN - Algorithm pairing

Question ID	Video A	Video B
1	SOTA	HUMAN
2	SANG	HUMAN
3	HUMAN	SANG
4	SOTA	HUMAN
5	HUMAN	SOTA
6	HUMAN	SANG
7	SOTA	HUMAN
8	HUMAN	SOTA
9	HUMAN	SANG
10	SANG	HUMAN

(b)

Questionnaire 1	Questionnaire 2
1. Navigation Turing Test on *SALSA-CPP* videos	**1.** Navigation Turing Test on *SALSA-PS* videos
2. Full PSI questionnaire on a video about the *SOTA algorithm* navigating in *SALSA-PS*	**2.** Full PSI questionnaire on a video about the *SANG algorithm* navigating in *SALSA-CPP*
3. Full PSI questionnaire on a video about the *SANG algorithm* navigating in *SALSA-PS*	**3.** Full PSI questionnaire on a video about the *SOTA algorithm* navigating in *SALSA-CPP*

Fig. 6.8 These figures outline the qualitative questionnaire's structure. (a) presents the SOTA and SANG algorithms' pairing with the human-generated (HUMAN) videos for the Navigation Turing Test (NTT) [35]. (b) presents the two created questionnaires' structure, detailing which algorithm – setting pairs were included.

Regarding the metrics of *Smoothness*, *Minimum distance to group centroids* and *Group O-space breach counts*, both SOTA and SANG algorithms failed to perform similarly to HUMAN baselines. However, SANG performed better with regards to *Smoothness* and *Minimum distance to group centroids*. The observed metric of *Group O-space breach counts* reflects that the SOTA algorithm was trained to avoid groups and their projected interaction circle, while SANG was only penalised if it entered a convex group space based on how the two algorithm's social interaction reward (r_t^{rs}) was formulated. Moreover, it can be seen that compared to the SOTA algorithm, SANG produced much smoother trajectories, meaning that it did not turn as much compared to the distance covered, which is more similar to how we humans would traverse crowds.

As for the qualitative metrics, I created two questionnaires (see Figure 6.8). One measured the Navigation Turing Test (NTT) [35] responses on videos collected from the SALSA Cocktail Party (SALSA-CPP) dataset. Then, it presented a video from the SOTA algorithm's navigation followed by the proposed SANG algorithm's navigation example, where the robot had to navigate the SALSA Poster Session (SALSA-PS) environment, and participants were required to answer *two* sets of Perceived Social Intelligence (PSI) questionnaires [11]. The second questionnaire began with an NTT about recordings from the SALSA-PS dataset and PSI questionnaires about videos collected with the SANG and SOTA algorithms navigating in the more narrow SALSA-CPP environment. As a result, I collected 24 and 14 responses to questionnaires *one* and *two*, respectively.

The Navigation Turing Test (NTT) [35] results can be seen in Table 6.3. It can be observed, that in the SALSA Poster Session (SALSA-PS) environment, people chose the SOTA algorithm 34.3% of the time as the human one, as opposed to the 32.8% of SANG. However, there was a much more drastic difference in the case of the SALSA Cocktail Party (SALSA-CPP) environment, where the pick ratio was 41.7% and 53.3% for the SOTA and SANG algorithms respectively. This means that in the SALSA-CPP

Table 6.3 Navigation Turing Test – This table presents the average percentage of people choosing an algorithm-generated trajectory over a human-generated one (Score), and the average confidence they had in this decision (Conf.) The table presents the comparison of the SOTA algorithm proposed by Do et al. [36], and my proposed algorithm, SANG. The test was performed on the SALSA Cocktail Party (SALSA-CPP) and Poster Session (SALSA-PS) datasets.

Test data	SALSA-PS		SALSA-CPP	
	Score	Conf.	Score	Conf.
SOTA [36]	**34.3%**$^{\circ}$	3.19	41.7%	3.34
SANG	32.8%	2.68	**53.3%**$^{\circ}$	3.09

$\circ$: significant ($p < .05$) difference when comparing the two methods in the two settings.

Table 6.4 Perceived Social Intelligence (PSI) Scores [11] – This table presents the measured PSI scores based on participants' answers. I compared the measured average (Avg.) and standard deviation (Std.) of Social Information Processing Total Scores (PSI-SIPT) and Social Presentation Total Scores (PSI-SPT) for both the SOTA algorithm [36] and my proposed solution, SANG. PSI-SIPT and PSI-SPT added up give the overall PSI score of the robot. The evaluation was done in the SALSA Poster Session (PS prefix) and Cocktail Party (CPP prefix) settings.

Score	PSI - SIPT		PSI - SPT	
	PS Avg. (Std.)	CPP Avg. (Std.)	PS Avg. (Std.)	CPP Avg. (Std.)
SOTA [36]	29.3 (7.91)	29.3 (6.95)	5.90 (3.77)	**5.94** (4.04)
SANG	**30.5** (7.39)	**30.6** (6.89)	**6.01** (3.42)	4.91 (4.44)

environment, SANG was picked to be the human-like solution over half of the time when compared against human-generated movements. Combining the data of both test settings, it can be observed that SANG was judged to be more human-like on average 43% of the time, significantly improving the SOTA's 38% chance based on an independent samples t-test [39], having t-score $= 2.64$ and p-value $= 0.0086$).

The Perceived Social Intelligence (PSI) Scores [11] (see Table 6.4) show that in all cases but the SALSA Poster Session's Social Presentation Total Scores (PSI-SPT), SANG performed better than the SOTA algorithm proposed by Do et al. [36]. The overall PSI scores can be calculated by adding up the PSI-SPT and Social Information Processing Total Scores (PSI-SIPT) scores, meaning that in the case of the SALSA Poster Session (SALSA-PS), the SOTA and SANG algorithms achieved scores of 35.2 and 36.5 respectively from the possible 66.5. Similarly, in the case of the SALSA Cocktail Party (SALSA-CPP) subset, these values were 35.2 and 35.0 for SOTA and SANG respectively. This means that while people perceived the SANG algorithm to be more capable of processing social

information, in the more crowded SALSA-CPP environment it was perceived to display worse social presentation skills.

I conjecture that the SOTA algorithm was chosen slightly more times in the more spacious, less crowded SALSA-PS environment as human-like based on the NTT tests as whenever it was possible, it had the opportunity to navigate around groups, which is considered to be the more socially acceptable behaviour. As for why it achieved better scores based on the PSI-SPT questionnaire in the case of the SALSA-PS dataset, I investigated the different components contributing to the Social Presentation Total score. The full list of PSI metrics, the measured values between the different SALSA settings, and two navigation approaches can be seen in Table 6.5. I observed that while all scores were similar in the SALSA-PS setting, and the *Friendly*, *Helpful*, *Caring*, *Trustworthy*, and *Conceited* ones remained similar in the SALSA-CPP setting, the robot guided by the proposed SANG algorithm was perceived as presenting more *Rude* (R) and *Hostile* (H) with scores of $R = -2$ and $H = -1.1$ as opposed to the SOTA algorithm's perceived $R = -1.7$ and $H = -0.8$. This might be due to the robot traversing the environment in a more goal-oriented manner instead of finding the small spaces between groups and navigating around them.

These results showcase that despite not performing to an equally socially-aware level, SANG improved compared to the SOTA by producing movement patterns with low smoothness which is a characteristic of human-generated data, comparable minimum distance to individuals, and more leniency for crossing groups when necessary, which trends the collected human benchmark also exhibited. These characteristics made SANG to be perceived as more capable of processing social information than the SOTA algorithm presented by Do et al. [36] and contributed to SANG being able to appear more human-like in the SALSA Cocktail Party (SALSA-CPP) setting based on the Navigation Turing Test results.

6.5 Discussion

To further improve the proposed solution, model stacking and/or transfer learning practices could be a promising research avenue. Namely, having a model learn to reach a goal in a static environment first, and in the second stage be tuned by introducing individuals and groups. This practice would have the potential to improve goal-reaching accuracy while maintaining the ability to perform more socially-aware navigation.

Moreover, to improve the training time required for the convergence of the RL algorithm, the implementation of an Asynchronous Advantage Actor-Critic (A3C) algorithm in the CrowdBot simulation environment [52] needs to be investigated.

Table 6.5 Detailed Perceived Social Intelligence (PSI) Scores [11] – This table presents the average measured PSI scores based on participants' answers. Social Information Processing Scores range from labels "RE" to "SOC", while Social Presentation Scores range from "FRD" to "HST". The table presents both the SOTA and SANG algorithm's results for both the SALSA Poster Session (SALSA-PS) and Cocktail Party (SALSA-CPP) settings.

Setting	PS		CPP	
Algorithm	SOTA	SANG	SOTA	SANG
Recognizes Human Emotions (RE)	**2.18**	2.16	2.09	**2.20**
Recognizes Human Behaviors (RB)	2.45	**2.66**	2.57	**2.63**
Recognizes Human Cognitions (RC)	1.91	**1.94**	**2.02**	1.91
Adapts to Human Emotions (AE)	1.80	**1.97**	1.84	**2.04**
Adapts to Human Behaviors (AB)	2.20	**2.33**	2.07	**2.14**
Adapts to Human Cognitions (AC)	2.18	**2.23**	2.09	**2.29**
Predicts Human Emotions (PE)	2.48	**2.53**	**2.46**	2.38
Predicts Human Behaviors (PB)	2.47	**2.50**	2.34	**2.55**
Predicts Human Cognitions (PC)	2.33	**2.42**	2.39	**2.46**
Identifies Humans (IH)	2.95	**3.03**	3.21	**3.30**
Identifies Individuals (II)	2.20	**2.43**	**2.05**	1.96
Identifies Social Groups (IG)	2.15	**2.29**	2.18	**2.21**
Social Competence (SOC)	**2.11**	2.09	**2.05**	2.02
Friendly (FRD)	2.45	**2.46**	**2.39**	2.38
Helpful (HLP)	2.40	**2.53**	**2.34**	2.32
Caring (CAR)	2.08	**2.15**	**2.25**	2.14
Trustworthy (TRU)	2.77	**2.78**	**2.77**	2.70
Rude (RUD) R	-1.82	**-1.80**	**-1.70**	-2.00
Conceited (CON) R	**-1.26**	-1.33	**-1.29**	-1.45
Hostile (HST) R	**-0.71**	-0.77	**-0.82**	-1.18

An additional consideration is the calculation of the social interaction reward defined in Eq. 5.1. Instead of assuming a noise-free reading, the reward could be continuous to instead account for how close the robot is to a group and reward it accordingly. Alternatively, instead of using a decision, the robot could learn this reward via a differential function. Furthermore, based on the qualitative evaluation results, a robot approaching groups at a certain distance may be perceived as socially aware in a densely crowded setting such as the SALSA-CPP test environment, but the same trajectory may be perceived as rude in the less crowded SALSA-PS setting. Therefore, the social interaction reward would benefit from crowd density being incorporated into a continuous function, this way establishing a trade-off between social gracefulness and efficient navigation.

Moreover, minimum distance to group boundaries should be investigated as a viable input feature instead of using group centroids as one of the input features when using convex group boundaries as a social reward. This information should prove to be more useful for the optimisation of the navigation algorithm, as it has a closer relation to the social group reward score.

With regards to evaluation results, it could be observed that the social reward function proposed by the SOTA is less forgiving, and is therefore more suitable for less densely crowded settings (e.g., SALSA-PS) in terms of qualitative metrics. Conversely, SANG performs better in a more dense environment such as the SALSA-CPP test setting. A possible reason behind this difference is that in a more crowded environment passing closer to groups or crossing them is more acceptable than in an environment where there is ample space between groups. Conversely, due to the formulation of the social interaction reward, the SOTA algorithm passes around groups at a larger distance (i.e., outside the perimeter of the group's circle used in reward calculation), which makes it seem more socially acceptable in the SALSA-PS setting. However, the same behaviour results in inefficient and not socially intelligent navigation in the more crowded SALSA-CPP environment.

The PSI questionnaire [11] used in the evaluation does not explicitly measure familiarity, predictability, or trust. Since the primary objective of implementing socially-aware navigation is to establish these characteristics, the qualitative assessment should be revised or extended to enable a more accurate and thorough measurement of these metrics.

The main focus of this study was to evaluate how group information can be leveraged to create socially-aware navigation in a real-to-sim environment. However, it is important to note that the study did not investigate the impact of group detection accuracy, nor did it assess the solution's robustness to errors in detection or noisy input features. To be deployed in the real world, a sensitivity analysis should be conducted to ensure the model remains reliable, even in noisy environments.

6.6 Conclusion

This work presented a novel socially-aware navigation algorithm based on an Advantage Actor-Critic (A2C) architecture. It used a more realistic social interaction reward score to represent the robot's relation to interaction groups and their boundaries. Moreover, it unified the evaluation practices in the field of socially-aware navigation, utilising a range of commonly used quantitative metrics, a qualitative measurement as well as a Navigation Turing Test (NTT) [35] which has not been applied to a real-to-sim context before.

The present results show that in several aspects, the proposed Socially-Aware Navigation between Groups (SANG) algorithm outperforms the previous state-of-the-art (SOTA) navigation approach presented by Do et al. [36] when it comes to social factors. The most notable difference is that the smoothness of the navigation was drastically lowered, which – according to the comparison with human benchmarks – is more comparable to how humans would navigate an environment. Moreover, the algorithm no longer refrains from crossing groups when necessary, which was possible with the previous SOTA, but in that case, the robot was punished for approaching vaguely defined group boundaries. This work utilised two qualitative approaches to measuring how well the robot performed from a social perspective. Based on the NTT, with the SANG algorithm, the robot was capable of performing navigation which would occasionally seem more human to observers than human-generated movements. Moreover, its perceived social skills were also improved compared to the SOTA algorithm's measured scores based on 3 of 4 Perceived Social Intelligence (PSI) Score [11] measurements.

In conclusion, the better handling of group data provided more valuable input to the A2C algorithm, and the collected evaluation methods provide a more thorough overview of how well the algorithms perform with regard to social awareness.

Chapter 7

Conclusions and Future Work

This thesis presented different aspects of a problem area concerning mobile robots' existence in human spaces and the difficulties arising from attempting to solve problems within this area purely from a robocentric perspective. The target research area within this field was how socially-aware navigation can be achieved in an indoor crowded environment, where the robot needs to traverse a crowd as individuals might require some form of help. The motivation behind this is that socially compliant robots fit more naturally into our environments, leading to their swift integration into our spaces. This work considered how the incorporation of social interaction groups can contribute to socially-aware navigation and detailed how such groups can be detected, and how their information can be used for navigation.

The literature overview presented in Chapter 2 indicated that existing solutions for group detection were not tested on robocentric datasets before, even though the shift in the point of view introduces several noise factors to the monitored features. Similarly, socially-aware navigation seldom used group information as a feature to achieve reliable navigation.

7.1 Summary of contributions

Based on the above-mentioned limitations, this thesis presented the following contributions.

RICA dataset

In Chapter 3, I introduced a novel robocentric dataset titled Robocentric Indoor Crowd Analysis (RICA) dataset, which captures an indoor reception setting. The dataset contains both annotated and unlabelled data. The annotations were done on a quarter of over an hour-long data with respect to both person- and group-level labels. Several practices

from other state-of-the-art third-person datasets were followed, most notably the validity checking by measuring inter-annotator agreement. Moreover, before this corpus no group datasets labelled F-formation types as defined by Kendon [75]. Lastly, being published before the work of Taylor et al [134] and Ehsanpour et al. [37], the RICA dataset was the first of its kind to introduce a robocentric dataset created specifically for group detection.

AHC based group detection

In Chapter 4, I presented an unsupervised technique based on the work of Japar et al. [70]. This work specifically targeted unsupervised approaches, as they have not been tested on robocentric datasets before. After testing the state-of-the-art Agglomerative Hierarchical Clustering (AHC) based method, I observed that only corner, width, height, and centroid information of identified people's bounding boxes will not yield optimal results when applied to a robot's camera feed. To improve the state-of-the-art, I introduced depth information and feature normalisation to the feature vectors proposed by Japar et al. [70]. Based on my results, these changes proved to be sufficient to improve unsupervised detection, however, the reliability of the solution remained strongly bottlenecked by the accuracy of human detectors.

GROWL and iGROWL

In Chapter 5, I introduced a Graph Neural Network based supervised technique for group detection. GROup detection With Link prediction (GROWL) was developed based on the core idea observed in the literature, creating pairwise affinity matrices to achieve group detection, and it was the first GNN based technique introduced for this problem. The advantage of this approach is that it utilises the inherent spatial layout of groups in a top-down representation of a scene. This work also described how robocentric data can be transformed into a graph representation. I tested GROWL against a state-of-the-art approach [60], a widely used group detection benchmark [124], and a later published GNN based solution [135], producing the best accuracy based on F_1-scores. However, in some cases it was prone to detecting only partial groups; e.g., a *two* and a *three*-person group instead of one with *five* people. To address this, I explored sample balancing techniques which can be applied to GNNs and proposed Improved GROup detection With Link prediction (iGROWL) as the new state-of-the-art solution.

SANG

In Chapter 6, I investigated the effect of using group information for improving socially-aware robot navigation. Previous approaches either did not use this feature or used more

abstract group information for their solutions. Following the state-of-the-art method proposed by Do et al. [36] I created an Advantage Actor-Critic (A2C) based solution titled Socially-Aware Navigation between Groups (SANG) to learn navigation behaviour incorporating a newly proposed social reward score based on convex group boundaries as opposed to centroid-only or abstract circular representations. My solution exhibited behaviour in two real-to-sim settings that was closer to how humans would navigate in these environments when compared with the previous state-of-the-art method.

Evaluation of socially-aware navigation

Previously, other solutions were not unified with regards to evaluation practices concerning socially-aware navigation and only compared algorithms to each other. However, I proposed an evaluation protocol incorporating both quantitative and qualitative metrics in a real-to-sim environment as opposed to the commonly used artificially generated simulation settings. Moreover, I incorporated an evaluation technique titled Navigation Turing Test [35] which I also applied to this context for a more thorough comparison. Finally, I evaluated my solution against both human-generated behaviours and another state-of-the-art technique, which was not commonplace in previous works that only compared quantitative and/or qualitative metrics between different algorithms, without establishing human benchmarks.

7.2 Future research directions

The following Sections present possible future directions based on the findings of the work presented in this thesis.

7.2.1 Robocentric group detection

This Section details both the supervised and unsupervised improvement possibilities identified throughout the literature review and the evaluations of the presented methods. Furthermore, it presents directions which would be beneficial to investigate regardless of which type of algorithm is used.

Agglomerative Hierarchical Clustering

My AHC based method could potentially be improved with other features. For example, an orientation feature could be introduced, which is also an observed input in other, unsupervised and supervised approaches. Computing orientation features in a robocentric

setting would be possible based on depth information by using the Deep-orientation algorithm [85].

iGROWL

The iGROWL algorithm could be further improved by introducing a self-supervising method to determine weak and strong edges in the graph and using this information to identify detected groups.

Moreover, there was no measurement done of the speed at which iGROWL can detect groups. Since one of the arguments for using unsupervised methods is that they sacrifice accuracy for speed, a comparative analysis could be conducted to measure how the two types of methods fare against each other. This is important, as based on the research and evaluation conducted on socially-aware navigation methods, group information is a valuable feature, but a mobile robot also needs to be able to produce actions in a short amount of time.

Common gaps

While unsupervised methods do not need labels to be trained, they still require human bounding boxes or position and orientation features as inputs. While human detection achieves good accuracy on third-person view datasets, robocentric datasets pose a challenge for most detectors due to occlusion, light condition changes, and other sources of noise. Therefore, to build a complete pipeline for a socially-aware robot, reliable, generally applicable techniques should be investigated, which can perform quick and accurate human detection on images taken from the robot's perspective.

Another avenue to be pursued is the additional information that can be obtained from detected groups. Barua et al. [14] proved in their work that knowing the F-formation type of detected groups is valuable information when a robot is required to navigate around and join a group. Based on their findings, it can be hypothesised that socially-aware navigation can be further improved if the groups are not only represented more accurately as investigated in Chapter 6 but their spatial alignment is also introduced in the feature vector for the input layer of the navigation algorithm.

Chapters 4 and 5, highlight the challenge of inaccurate human detections, which can hinder the performance of group detection approaches. Furthermore, inaccurate group detection may affect socially-aware navigation if it is used as a feature. Apart from performing tests to measure how robust these models are to human detection errors, an alternative approach is to develop an end-to-end system that takes sensor information as input and directly produces group detections or navigation commands without intermediate steps. However, in this dissertation, I decided to investigate individual models instead of an

end-to-end approach. I found the explicit use of graphs and group information particularly useful for the group detection and navigation tasks, respectively. Moreover, the methods covered in this thesis provide an anthropomorphic approach to how the robot should process raw information in order to create a human-like end result. Finally, end-to-end systems pose a different array of problems, particularly, they might fail, and it would be challenging to investigate why.

7.2.2 Socially-aware navigation

This Section presents possibilities for improving approaches in socially-aware navigation as well as specific improvements with regard to the simulation environment in real-to-sim settings and how the evaluation could be further improved.

Navigation approaches

Socially-aware navigation approaches could be further improved if more social features were introduced to them, making the robots more understanding of the social context. This is an ability that, based on the Perceived Social Intelligence (PSI) questionnaire [11] based evaluation presented in Chapter 6, many people did not observe when looking at the developed algorithms navigating social settings. Evidently, the social context of groups aids robots in navigating in a manner that's perceived as more natural, but as observed, the way group information is represented has a significant effect on the outcome.

One of the ways in which socially-aware navigation could be improved is, in the case of Reinforcement Learning algorithms, by computing group based social rewards based on the robot's distance from the convex space formed by participants. Treating groups as convex social entities would allow more space for robots to move in between them, especially in very crowded spaces, and would reflect disturbing the interaction better in general compared to group centroid or circular group representation based reward calculations.

Moreover, from the group detection stage of an autonomous navigation solution, robots could utilise F-formation labels, as investigated by Barua et al. [14], to navigate between groups. This information will also bear significance in case the robot's goal will not be a specific location in a setting, but to approach a single person or a group. In such cases, mobile robots need to know how to approach individuals who might or might not be engaged in interaction within a group, or merge into groups if their task involves doing so.

Based on the work of Thompson et al. [135] who incorporated acceleration features as inputs for their GNN based approach, velocities of the robot, individuals, or groups in an environment could be a viable input feature for navigation algorithms. Similarly, following several works on trajectory prediction in crowds, taking more than a single timestamp's

information could prove to be beneficial for the improvement of navigation in general, especially in more dynamic crowds.

The work described in Chapter 6 only used GT data and noise-free features to navigate the environment in order to compare to the SOTA. However, in future work, a sensitivity analysis should be conducted, which measures the algorithm's robustness to noise and imperfect group detections in particular. Furthermore, the social reward score should also be improved to accommodate more realistic, continuous boundaries that allow for a margin of error and incorporates crowd density in its calculation. These two considerations would be necessary for the more reliable real-life deployment of the socially-aware navigation approach.

Simulation environment

The community would benefit from an improved simulation environment. While the CrowdBot [52] simulation is sufficient for the evaluation of socially-aware navigation solutions, its structure is not fit for the training of machine learning, let alone Reinforcement Learning based solutions. CrowdBot itself is a useful tool for the creation of real-life dataset based real-to-sim environments, but its single simulated environment does not allow asynchronous training of a network. Therefore, it could be improved by integrating it into the multi-threaded AI training ecosystem named ML-Agents toolkit, which is provided by Unity3D [55].

The simulation environment would also benefit from having a more recent ROS [116] backend like 'noetic', or even the more recent ROS2 system, which would allow the integration of the latest machine learning techniques which all use *python3*.

Evaluation method

Even though a human benchmark is beneficial for the evaluation of socially-aware navigation solutions, due to the different perspectives of human controllers, their human-generated benchmarks are not completely accurate. By viewing the environment they need to traverse from a third-person view, they have a better overview of both the location of the goal and the groups in the environment. Therefore, they might find it easier to determine the best path between groups to execute during navigation. A solution to this perspective mismatch could be to control the robot from a first-person view or to create a more immersive experience in the real-to-sim setting. This could be achieved by making simulated agents appear more realistically, and placing navigating human participants into the environment via virtual reality.

Another improvement targeting specifically the Perceived Social Intelligence (PSI) questionnaires [11] for qualitative assessment would be to revise the 20 categories involved

in the calculation, especially the Social Information Processing Scores. Based on feedback collected at the end of the questionnaires presented in Chapter 6, participants often highlighted that questions related to the robot's *recognition* (*R_*), *adaptation* (*A_*) to, or *prediction* (*P_*) of human *emotions* (*_E*) and *cognitions* (*_C*) are hard to answer. These metrics are also not strongly relevant to the measurement of socially-aware navigation. However, their omission would make the solution incomparable to other techniques striving to achieve good Perceived Social Intelligence.

References

[1] Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1.

[2] AbuOda, G., De Francisci Morales, G., and Aboulnaga, A. (2020). Link prediction via higher-order motif features. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, pages 412–429. Springer.

[3] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

[4] Al Shalabi, L., Shaaban, Z., and Kasasbeh, B. (2006). Data mining: A preprocessing engine. *Journal of Computer Science*, 2(9):735–739.

[5] Alameda-Pineda, X., Staiano, J., Subramanian, R., Batrinca, L., Ricci, E., Lepri, B., Lanz, O., and Sebe, N. (2016). Salsa: A novel dataset for multimodal group behavior analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1707–1720.

[6] Alletto, S., Serra, G., Calderara, S., and Cucchiara, R. (2015). Understanding social relationships in egocentric vision. *Pattern Recognition*, 48(12):4082–4096.

[7] Alletto, S., Serra, G., Calderara, S., Solera, F., and Cucchiara, R. (2014). From ego to nos-vision: Detecting social relationships in first-person views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 580–585.

[8] Alsentzer, E., Finlayson, S., Li, M., and Zitnik, M. (2020). Subgraph neural networks. *Advances in Neural Information Processing Systems*, 33:8017–8029.

[9] Antonakos, E., Alabort-i Medina, J., and Zafeiriou, S. (2015). Active pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5435–5444.

[10] Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. *Machine learning*, 56(1):89–113.

[11] Barchard, K. A., Lapping-Carr, L., Westfall, R. S., Banisetty, S. B., and Feil-Seifer, D. (2018). Perceived social intelligence (psi) scales test manual. *Unpublished psychological test and test manual. Observer report of*, 20.

[12] Bartneck, C., Kulic, D., and Croft, E. (2008). Measuring the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *Metrics for Human-Robot Interaction 2008*, page 37.

[13] Bartneck, C., Kulić, D., Croft, E., and Zoghbi, S. (2009). Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1:71–81.

[14] Barua, H. B., Pramanick, P., Sarkar, C., and Mg, T. H. (2020). Let me join you! real-time f-formation recognition by a socially aware robot. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 371–377.

[15] Bazzani, L., Cristani, M., and Murino, V. (2012). Decentralized particle filter for joint individual-group tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1886–1893. IEEE.

[16] Bazzani, L., Cristani, M., Tosato, D., Farenzena, M., Paggetti, G., Menegaz, G., and Murino, V. (2013). Social interactions by visual focus of attention in a three-dimensional environment. *Expert Systems*, 30(2):115–127.

[17] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

[18] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[19] Cabrera-Quiros, L., Demetriou, A., Gedik, E., van der Meij, L., and Hung, H. (2018). The matchnmingle dataset: a novel multi-sensor resource for the analysis of social interactions and group dynamics in-the-wild during free-standing conversations and speed dates. *IEEE Transactions on Affective Computing*, 12(1):113–130.

[20] Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

[21] Carmeli, C., Knyazeva, M. G., Innocenti, G. M., and De Feo, O. (2005). Assessment of eeg synchronization based on state-space analysis. *Neuroimage*, 25(2):339–354.

[22] Carpinella, C. M., Wyman, A. B., Perez, M. A., and Stroessner, S. J. (2017). The robotic social attributes scale (rosas) development and validation. In *Proceedings of the 2017 ACM/IEEE International Conference on human-robot interaction*, pages 254–262.

[23] Chandran, A. K., Poh, L. A., and Vadakkepat, P. (2015). Identifying social groups in pedestrian crowd videos. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6. IEEE.

[24] Chen, M., Wang, Q., and Li, X. (2017a). Anchor-based group detection in crowd scenes. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1378–1382. IEEE.

[25] Chen, Y. F., Everett, M., Liu, M., and How, J. P. (2017b). Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1343–1350. IEEE.

[26] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

[27] Coleman, J. S. and James, J. (1961). The equilibrium size distribution of freely-forming groups. *Sociometry*, 24(1):36–45.

[28] Cooper, S., Di Fava, A., Vivas, C., Marchionni, L., and Ferro, F. (2020). Ari: The social assistive robot and companion. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 745–751. IEEE.

[29] Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

[30] Cristani, M. (2012). Decentralized particle filter for joint individual-group tracking. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, page 1886–1893, USA. IEEE Computer Society.

[31] Cristani, M., Bazzani, L., Paggetti, G., Fossati, A., Tosato, D., Del Bue, A., Menegaz, G., and Murino, V. (2011). Social interaction discovery by statistical analysis of f-formations. In *BMVC*, volume 2, pages 10–5244.

[32] Cybenko, G., O'Leary, D. P., and Rissanen, J. (1998). *The Mathematics of Information Coding, Extraction and Distribution*, volume 107. Springer Science & Business Media.

[33] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

[34] Davies, D. and Bouldin, D. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, 1(2):224–227.

[35] Devlin, S., Georgescu, R., Momennejad, I., Rzepecki, J., Zuniga, E., Costello, G., Leroy, G., Shaw, A., and Hofmann, K. (2021). Navigation turing test (ntt): Learning to evaluate human-like navigation. In *International Conference on Machine Learning*, pages 2644–2653. PMLR.

[36] Do, N. T., Pham, T. D., Son, N. H., Ngo, T. D., and Truong, X. T. (2020). Deep reinforcement learning based socially aware mobile robot navigation framework. In *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 226–231. IEEE.

[37] Ehsanpour, M., Saleh, F., Savarese, S., Reid, I., and Rezatofighi, H. (2022). Jrdb-act: A large-scale dataset for spatio-temporal action, social group and activity detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20983–20992.

[38] Elassal, N. and Elder, J. H. (2017). Unsupervised crowd counting. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pages 329–345. Springer.

[39] Everitt, B. and Skrondal, A. (2010). *The Cambridge Dictionary of Statistics*. Cambridge University Press.

[40] Faisal, M., Al-Mutib, K., Hedjar, R., Mathkour, H., Alsulaiman, M., and Mattar, E. (2013). Multi modules fuzzy logic for mobile robots navigation and obstacle avoidance in unknown indoor dynamic environment. In *Proceedings of the 2013 International Conference on Systems, Control and Informatics*, pages 371–379.

[41] Fathi, A., Hodgins, J. K., and Rehg, J. M. (2012). Social interactions: A first-person perspective. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1226–1233. IEEE.

[42] Ferryman, J. and Shahrokni, A. (2009). Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–6. IEEE.

[43] Field, T., Leibs, J., Bowman, J., Thomas, D., and Perron, J. (2020). rosbag - ROS Wiki. http://wiki.ros.org/rosbag. original-date: 2020-06-11.

[44] Finley, T. and Joachims, T. (2005). Supervised clustering with support vector machines. In *Proceedings of the 22nd international conference on Machine learning*, pages 217–224.

[45] Fruin, J. J. (1970). *Designing for pedestrians a level of service concept*. Polytechnic University.

[46] Gandhi, D., Pinto, L., and Gupta, A. (2017). Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955. IEEE.

[47] Gao, Y. and Huang, C.-M. (2022). Evaluation of socially-aware robot navigation. *Frontiers in Robotics and AI*, 8:420.

[48] Ge, W., Collins, R. T., and Ruback, R. B. (2012). Vision-based analysis of small groups in pedestrian crowds. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):1003–1016.

[49] Gedik, E. and Hung, H. (2016). Speaking status detection from body movements using transductive parameter transfer. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing: adjunct*, pages 69–72.

[50] Gerkey, B., Vaughan, R. T., Howard, A., et al. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, volume 1, pages 317–323. Citeseer.

[51] Goldberg, L. R. (1992). The development of markers for the big-five factor structure. *Psychological assessment*, 4(1):26.

[52] Grzeskowiak, F., Gonon, D., Dugas, D., Paez-Granados, D., Chung, J. J., Nieto, J., Siegwart, R., Billard, A., Babel, M., and Pettré, J. (2021). Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3879–3885. IEEE.

[53] Guerry, J., Wolf, C., Lombardi, E., and Mille, J. (2017). actanno-2. https://github.com/jorisgu/actanno-2. original-date: 2017-07-06.

[54] Guizzo, E. (2019). By leaps and bounds: An exclusive look at how boston dynamics is redefining robot agility. *IEEE Spectrum*, 56(12):34–39.

[55] Haas, J. K. (2014). A history of the unity game engine. *Diss. Worcester Polytechnic Institute*, 483(2014):484.

[56] Hall, E. T., Birdwhistell, R. L., Bock, B., Bohannan, P., Diebold Jr, A. R., Durbin, M., Edmonson, M. S., Fischer, J., Hymes, D., Kimball, S. T., et al. (1968). Proxemics [and comments and replies]. *Current anthropology*, 9(2/3):83–108.

[57] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[58] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

[59] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning. *Image Recognition*, 7.

[60] Hedayati, H., Muehlbradt, A., Szafir, D. J., and Andrist, S. (2020). Reform: Recognizing f-formations for social robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11181–11188. IEEE.

[61] Hedayati, H., Szafir, D., and Andrist, S. (2019). Recognizing f-formations in the open world. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 558–559. IEEE.

[62] Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282.

[63] Honour, A., Banisetty, S. B., and Feil-Seifer, D. (2021). Perceived social intelligence as evaluation of socially navigation. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pages 519–523.

[64] Hu, P. and Ramanan, D. (2017). Finding tiny faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 951–959.

[65] Huang, J., Ertekin, S., and Giles, C. L. (2006). Efficient name disambiguation for large-scale databases. In *European conference on principles of data mining and knowledge discovery*, pages 536–544. Springer.

[66] Huang, J., Tan, D., Sun, L., Shao, J., Ma, Y., and Wang, Z. (2018). Obstacle recognition in front of vehicle based on geometry information and corrected laser intensity. *Artificial Life and Robotics*, 23(3):338–344.

[67] Hung, H. and Ba, S. O. (2009). Speech/non-speech detection in meetings from automatically extracted low resolution visual features. Technical report, Idiap.

[68] Hung, H. and Kröse, B. (2011). Detecting f-formations as dominant sets. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 231–238.

[69] Inaba, S. and Aoki, Y. (2016). Conversational group detection based on social context using graph clustering algorithm. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 526–531. IEEE.

[70] Japar, N., Chan, C. S., and Kok, V. J. (2019). Coherent crowd analysis in still image. In *2019 IEEE 21st international workshop on multimedia signal processing (MMSP)*, pages 1–6. IEEE.

[71] Jiang, Y., Yang, Z., Guo, J., Li, H., Liu, Y., Guo, Y., Li, M., and Pu, X. (2021). Coupling complementary strategy to flexible graph neural network for quick discovery of coformer in diverse co-crystal materials. *Nature Communications*, 12(1):5950.

[72] Kamil, F., Tang, S., Khaksar, W., Zulkifli, N., and Ahmad, S. (2015). A review on motion planning and obstacle avoidance approaches in dynamic environments. *Advances in Robotics & Automation*, 4(2):134–142.

[73] Katyal, K., Gao, Y., Markowitz, J., Pohland, S., Rivera, C., Wang, I.-J., and Huang, C.-M. (2022a). Learning a group-aware policy for robot navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11328–11335. IEEE.

[74] Katyal, K., Gao, Y., Markowitz, J., Pohland, S., Rivera, C., Wang, I.-J., and Huang, C.-M. (2022b). Learning a group-aware policy for robot navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11328–11335.

[75] Kendon, A. (1990). *Conducting interaction: Patterns of behavior in focused encounters*. Conducting interaction: Patterns of behavior in focused encounters. Cambridge University Press, New York, NY, US.

[76] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[77] Koos, S., Mouret, J.-B., and Doncieux, S. (2010). Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 119–126.

[78] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

[79] Kretzschmar, H., Spies, M., Sprunk, C., and Burgard, W. (2016). Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307.

[80] Kruse, T., Pandey, A. K., Alami, R., and Kirsch, A. (2013). Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743.

[81] Ladickỳ, L., Russell, C., Kohli, P., and Torr, P. H. (2013). Inference methods for crfs with co-occurrence statistics. *International journal of computer vision*, 103(2):213–225.

[82] Large, F., Laugier, C., and Shiller, Z. (2005). Navigation among moving obstacles using the NLVO: Principles and applications to intelligent vehicles. *Autonomous Robots*, 19(2):159–171.

[83] LeCun, Y., Muller, U., Ben, J., Cosatto, E., and Flepp, B. (2005). Off-road obstacle avoidance through end-to-end learning. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 739–746. MIT Press.

[84] Lerner, A., Chrysanthou, Y., and Lischinski, D. (2007). Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library.

[85] Lewandowski, B., Seichter, D., Wengefeld, T., Pfennig, L., Drumm, H., and Gross, H.-M. (2019). Deep orientation: Fast and robust upper body orientation estimation for mobile robotic applications. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 441–448. IEEE.

[86] Liang, X., Shen, X., Feng, J., Lin, L., and Yan, S. (2016). Semantic object parsing with graph lstm.

[87] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

[88] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

[89] Liu, S., Chang, P., Huang, Z., Chakraborty, N., Hong, K., Liang, W., McPherson, D. L., Geng, J., and Driggs-Campbell, K. (2022). Intention aware robot crowd navigation with attention-based interaction graph.

[90] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer.

[91] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

[92] Lund, H. H. and Miglino, O. (1996). From simulated to real robots. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 362–365. IEEE.

[93] Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer Research*, 27(2):209–220.

[94] Marshall, P., Rogers, Y., and Pantidi, N. (2011). Using f-formations to analyse spatial patterns of interaction in physical environments. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 445–454.

[95] Martín-Martín, R., Rezatofighi, H., Shenoi, A., Patel, M., Gwak, J., Dass, N., Federman, A., Goebel, P., and Savarese, S. (2019). JRDB: A dataset and benchmark for visual perception for navigation in human environments. *CoRR*, abs/1910.11792.

[96] Mavrogiannis, C. I., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., and Oh, J. (2021). Core challenges of social robot navigation: A survey. *CoRR*, abs/2103.05668.

[97] Mavrogiannis, C. I., Blukis, V., and Knepper, R. A. (2017). Socially competent navigation planning by deep learning of multi-agent path topologies. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6817–6824. IEEE.

[98] Mazzon, R., Poiesi, F., and Cavallaro, A. (2013). Detection and tracking of groups in crowd. *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 202–207.

[99] Milde, M. B., Blum, H., Dietmüller, A., Sumislawska, D., Conradt, J., Indiveri, G., and Sandamirskaya, Y. (2017). Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system. *Frontiers in neurorobotics*, 11:28.

[100] Minguez, J. and Montano, L. (2004). Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59.

[101] Mirsky, R., Xiao, X., Hart, J., and Stone, P. (2021). Prevention and resolution of conflicts in social navigation–a survey. *arXiv preprint arXiv:2106.12113*.

[102] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR.

[103] Moor, J. H. (2003). Turing test. In *Encyclopedia of Computer Science*, pages 1801–1802.

[104] Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R., and Yuille, A. (2014). The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898.

[105] Moussaïd, M., Perozo, N., Garnier, S., Helbing, D., and Theraulaz, G. (2010). The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4):e10047.

[106] My, C. A., Truong, X. T., et al. (2020). Toward socially aware trajectory planning system for autonomous mobile robots in complex environments. In *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 90–95. IEEE.

[107] Nasrinahar, A. and Chuah, J. H. (2018). Intelligent motion planning of a mobile robot with dynamic obstacle avoidance. *Journal on Vehicle Routing Algorithms*, 1:89–104.

[108] Ostendorff, M., Rethmeier, N., Augenstein, I., Gipp, B., and Rehm, G. (2022). Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11670–11688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[109] Pages, J., Marchionni, L., and Ferro, F. (2016). Tiago: the modular robot that adapts to different research needs. In *International workshop on robot modularity, IROS*, volume 290.

[110] Pandey, A. K., Gelin, R., and Robot, A. (2018). Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine*, 25(3):40–48.

[111] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[112] Pavan, M. and Pelillo, M. (2006). Dominant sets and pairwise clustering. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):167–172.

[113] Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE.

[114] Pels, F., Kleinert, J., and Mennigen, F. (2018). Group flow: A scoping review of definitions, theoretical approaches, measures and findings. *PloS one*, 13(12):e0210117.

[115] PMO, C. (2020). Crowdbot - safe robot navigation in dense crowds. https://crowdbot.eu/wp-content/uploads/2020/05/CROWDBOT-Project-D73_CROWDBOT-Challenge_v2.pdf. original-date: 2020-05-06.

[116] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. (2009). Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan.

[117] Ramírez, O. A. I., Varni, G., Andries, M., Chetouani, M., and Chatila, R. (2016). Modeling the dynamics of individual behaviors for group detection in crowds using low-level features. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1104–1111. IEEE.

[118] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

[119] Rios-Martinez, J., Spalanzani, A., and Laugier, C. (2015). From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics*, 7(2):137–153.

[120] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

[121] Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

[122] Schmuck, V. (2020). actanno-v3. https://github.com/d4rkspir1t/actanno-v3. original-date: 2020-01-03.

[123] Schmuck, V. and Meredith, D. (2019). Training networks separately on static and dynamic obstacles improves collision avoidance during indoor robot navigation. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 655–660. ESANN.

[124] Setti, F., Russell, C., Bassetti, C., and Cristani, M. (2015). F-formation detection: Individuating free-standing conversational groups in images. *PloS one*, 10(5):e0123783.

[125] Shaikh, M. B. and Chai, D. (2021). Rgb-d data-based action recognition: A review. *Sensors*, 21(12):4246.

[126] Shiller, Z., Large, F., and Sekhavat, S. (2001). Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 3716–3721. IEEE.

[127] Shiomi, M., Zanlungo, F., Hayashi, K., and Kanda, T. (2014). Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model. *International Journal of Social Robotics*, 6(3):443–455.

[128] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[129] Smith, K., Gatica-Perez, D., Odobez, J.-M., and Ba, S. (2005). Evaluating multi-object tracking. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)-workshops*, pages 36–36. IEEE.

[130] Sokal, R. R. and Michener, C. D. (1975). A statistical method for evaluating systematic relationships. *Multivariate statistical methods, among-groups covariation*, page 269.

[131] Stone, C. J. (1977). Consistent nonparametric regression. *The annals of statistics*, pages 595–620.

[132] Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

[133] Tan, S., Tax, D. M., and Hung, H. (2022). Conversation group detection with spatio-temporal context. *arXiv preprint arXiv:2206.02559*.

[134] Taylor, A., Chan, D. M., and Riek, L. D. (2020). Robot-centric perception of human groups. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(3):1–21.

[135] Thompson, S., Gupta, A., Gupta, A. W., Chen, A., and Vázquez, M. (2021). Conversational group detection with graph neural networks. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 248–252.

[136] Truong, X.-T. and Ngo, T.-D. (2016). Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. *International Journal of Social Robotics*, 8:663–684.

[137] Truong, X.-T. and Ngo, T.-D. (2018). "to approach humans?": A unified framework for approaching pose prediction and socially aware robot navigation. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):557–572.

[138] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104.

[139] Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer.

[140] Vascon, S., Mequanint, E. Z., Cristani, M., Hung, H., Pelillo, M., and Murino, V. (2015). A game-theoretic probabilistic approach for detecting conversational groups. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part V 12*, pages 658–675. Springer.

[141] Vasquez, D., Okal, B., and Arras, K. O. (2014). Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1341–1346. IEEE.

[142] Vechtomova, O. (2009). Book review: Introduction to information retrieval by christopher d. manning, prabhakar raghavan, and hinrich schütze. *Computational Linguistics*, 35(2).

[143] Vinciarelli, A., Pantic, M., and Bourlard, H. (2009a). Social signal processing: Survey of an emerging domain. *Image and vision computing*, 27(12):1743–1759.

[144] Vinciarelli, A., Pantic, M., and Bourlard, H. (2009b). Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759. Visual and multimodal analysis of human spontaneous behaviour:.

[145] Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080.

[146] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

[147] Wolf, C., Lombardi, E., Mille, J., Celiktutan, O., Jiu, M., Dogan, E., Eren, G., Baccouche, M., Dellandréa, E., Bichot, C.-E., et al. (2014). Evaluation of video activity localizations integrating quality and quantity measurements. *Computer Vision and Image Understanding*, 127:14–30.

[148] Yamamoto, T., Terada, K., Ochiai, A., Saito, F., Asahara, Y., and Murase, K. (2019). Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH journal*, 6(1):1–15.

[149] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 7444–7452.

[150] Yang, F., Gao, Y., Ma, R., Zojaji, S., Castellano, G., and Peters, C. (2021). A dataset of human and robot approach behaviors into small free-standing conversational groups. *PLOS ONE*, 16(2):1–24.

[151] Yang, S., Luo, P., Loy, C.-C., and Tang, X. (2016). Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533.

[152] Zen, G., Lepri, B., Ricci, E., and Lanz, O. (2010). Space speaks: towards socially and personality aware visual surveillance. In *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis*, pages 37–42.

[153] Zhang, L. and Hung, H. (2016). Beyond f-formations: Determining social involvement in free standing conversing groups from static images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1086–1095.

[154] Zhang, L. and Hung, H. (2018). On social involvement in mingling scenarios: Detecting associates of f-formations in still images. *IEEE Transactions on Affective Computing*, 12(1):165–176.

[155] Zhang, M. and Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31.

[156] Zhang, Y., Zhou, D., Chen, S., Gao, S., and Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597.

[157] Zhao, K., Zhou, L., Hu, Z., Cheng, S., Shi, A., Sun, Y., and Liu, J. (2021). Human-aware robot navigation based on asymmetric gaussian model. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 322–327. IEEE.

[158] Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, page 1433–1438. AAAI Press.