



**HAL**  
open science

## Group Abstraction for Assisted Navigation of Social Activities in Intelligent Environments

Thomas Given-Wilson, Axel Legay, Sean Sedwards, Olivier Zendra

► **To cite this version:**

Thomas Given-Wilson, Axel Legay, Sean Sedwards, Olivier Zendra. Group Abstraction for Assisted Navigation of Social Activities in Intelligent Environments. *Journal of Reliable Intelligent Environments*, 2018, 4 (2), pp.107-120. 10.1007/s40860-018-0058-1 . hal-01629137v2

**HAL Id: hal-01629137**

**<https://inria.hal.science/hal-01629137v2>**

Submitted on 5 Sep 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Group Abstraction for Assisted Navigation of Social Activities in Intelligent Environments

Thomas Given-Wilson · Axel Legay · Sean Sedwards · Olivier Zendra

Received: date / Accepted: date

**Abstract** The ACANTO project is developing robotic assistants to aid the confidence and recovery of older adults. A key requirement of these assistants is aiding with navigation in complex and potentially chaotic environments. Prior work has addressed this for a single user, using a single robotic assistant in an intelligent environment. However, for therapeutic purposes, ACANTO supports social groups and group activities. ACANTO’s robotic assistants must therefore be able to plan the motion of groups of older adults walking together. This requires an efficient navigation solution that can handle large numbers of users and that can operate rapidly on embedded computing devices. To increase user confidence, the solution must encourage group cohesion without trying to impose its own rigid structure; it must try to maintain the natural (de facto) group structure despite unpredictable behaviours and environmental conditions.

Our on-the-fly group motion planner addresses these challenges by: using intelligent environment information to develop behavioural traces, clustering traces to determine groups, constructing a predictive model of the groups as a whole, and finding an optimal suggested trajectory using Statistical Model Checking.

---

T. Given-Wilson · A. Legay · O. Zendra  
Inria, Campus universitaire de Beaulieu, Rennes 35000, France

S. Sedwards  
University of Waterloo, 200 University Avenue West, Waterloo,  
Ontario, Canada

This work is partly supported by the Japanese Science and Technology agency (JST) ERATO project JPMJER1603: HASUO Metamathematics for Systems Design.

In this work we describe our proposed approach in detail and validate some of its novel aspects on the ETH Zürich pedestrian motion dataset.

**Keywords** Assisted living · Intelligent environments · Confidence · Group motion planning · Therapeutic group activities

## 1 Introduction

The ACANTO EU project [1] is developing robotic assistants to improve the mobility and confidence of mobility-impaired and elderly adults. These assistants provide a variety of support to their users, including: navigational assistance, social networking, social activity planning, therapeutic regime support, and diagnostic support. As physical devices, the assistants are user-propelled “walkers”, equipped with an embedded computing device (e.g. a tablet computer), sensors (e.g. cameras), and capability to communicate with a server via a radio link.

Navigational assistance during social activities is the focus of this paper. The specific goal is to help users, engaged in a therapeutic group activity, navigate in a potentially busy dynamic environment while also maintaining social group cohesion. The intention of such activities is to improve physical mobility and develop social interactions, thus improving a user’s confidence in everyday activities.

A precursor robotic assistant has been developed before in the DALi project [12, 11], acting selfishly to ensure the safe navigation of a single user. The DALi robot assistant has on-board sensors to observe a limited field of view in front of the robot, using the information from these to provide navigational assistance to its user. This was achieved with a “predictor-corrector” architecture, comprising the *social force model* and *statistical model*

*checking* in a reactive planner that frequently replans and makes immediate navigational suggestions to the user. The key operational loop of this solution is to: observe the environment, model the agents in the environment in the social force model, and then use statistical model checking (given the safety constraints for the user) to find the optimal next move to suggest to the user. As with common vehicle navigation aids, the user is free to accept or reject the suggestion.

In ACANTO the robot assistant is now part of a larger intelligent environment of networked robots, servers, and sensors. In addition to the on-board sensors, each robot assistant can be sent further information gathered by other robot assistants and environmental sensors. This information is provided by a shared server infrastructure, allowing for efficient pruning of non-local information. In practice, this allows robot assistants to substantially improve their field of view and vision range, as well as have information about other members of a social group.

The generalisation from a single robot to groups of robot assistants that should maintain group cohesion poses several significant challenges. *Computationally*, the task is exponential in the number of users, considering all the possible combinations of their navigational choices. *Incomplete information* is normal, since sensors are distributed between robotic assistants and the environment, and communication may fail, leading to robots having different perceptions of the current global state of the system. *Maintaining group cohesion* is non-trivial, since group composition and position are dynamic and, unlike swarm robotics, no group member can be abandoned. *Frequent replanning* is necessary, since in the chaotic environment and without control of the robot, navigation and sensor information will rapidly become out-dated. This is further complicated by the robot assistant only being able to advise the user, and so navigational advice may be ignored.

Our solution is to *abstract the navigation task away from individuals in favour of groups of users*. This group abstraction is achieved by building behavioural *traces* and then grouping them by common characteristics. We then use the group abstraction to plan the motion of individuals in an efficient way.

The first step in our solution is for the robot assistant to obtain traces that provide past and present behavioural information about users and other pedestrians in the environment. These traces are extracted from available sensor information (gathered from on-board sensors as well as received from the ACANTO server infrastructure that has information from other robots and environmental sensors). Particular care must be taken when constructing traces, to account for incom-

plete sensor information and pedestrians that move in or out of sensor range. We present techniques to generate traces that can efficiently and accurately account for these challenges.

The second step is to cluster traces into groups of pedestrians with common behaviour. Traces generally contain an individual’s history of position and dynamics (velocity and acceleration), yielding groups whose members share more than simple instantaneous proximity. Group members therefore share *deep* behavioural similarity, potentially leading to models of crowded scenarios where groups moving in different ways interleave.

By treating the groups of pedestrians as social particles, we are able to re-use our efficient predictor-corrector navigational architecture. In this case, however, the structure and the parameters of the social force model (or some other predictive model of pedestrian behaviour) must be modified to account for group dynamics. Statistical model checking is used to find the probabilistically optimal “next move” for the group, which is then personalised for each member. This approach inherits the benefits of our previous work, while addressing many of the difficulties that arise from generalising to groups.

There are many alternative motion planning solutions in the literature, although none is without problems in our particular context. Of those that concern groups in some way, most do not include “humans in the loop” and tend to assume that the robots are fully cooperative with their planned motion. This is not our case. The approaches that do include humans, either do not consider groups, or fail to consider the other inherent uncertainties and requirements of ACANTO. These issues are covered more fully in Section 10.

In what follows we therefore motivate our choices and demonstrate the effectiveness of our approach by validating our group abstraction mechanisms on the ETH Zürich BIWI walking pedestrians dataset [15]. We thus show these mechanisms operate correctly and effectively on real world trajectories of pedestrians in a chaotic environment. In this particular case we are also able to improve on the original data annotations.

The rest of the paper is structured as follows. Section 2 recalls background technologies useful to understanding the sequel. Section 3 overviews the key aspects of the ACANTO project as the context for this work. Section 4 identifies the challenges to be addressed for robotic assistants. Section 5 overviews our presented solution. Section 6 details the algorithms used to identify and update traces within our reactive planner. Section 7 presents how we perform group abstraction on the traces, in order to identify social groups from trace behaviour. Section 8 shows how to combine these in an efficient reactive planner that fits within the ACANTO

architecture. Section 9 validates the algorithms using the ETH Zürich pedestrian data set. Section 10 contrasts our approach with related work. Section 11 concludes.

## 2 Background

The algorithms we present make use of versions of the social force model (SFM) to approximately predict the future behaviour of individual pedestrians and that of pedestrian groups, given specified immediate actions of the users of the system. The algorithms then employ statistical model checking (SMC) to find the actions which approximately maximise the probability of “success”. This section thus briefly describes the key concepts of the SFM and SMC.

### 2.1 The Social Force Model

The SFM [24,23,22,21] is a widely applied model of pedestrian motion, which represents the social and physical interactions of human agents as explicit forces. For example, the general desire of pedestrians not to collide with fixed objects and other pedestrians can be modelled by repulsive forces that keep them apart, while the desire of friends to walk together can be modelled as an attractive force between them. The limitations of the SFM have been well documented (e.g. [34]), but these are not critical to our application, nor is our approach dependent on using the SFM. We nevertheless present below the key elements of the standard SFM. For further discussion and applications of the SFM, including variations that account for more human-like motion, the reader is referred to [16,17] and Section 10.3.

Denoting vectors in bold type, agent  $i$  has mass  $m_i$  centered at position  $\mathbf{x}_i \in \mathbb{R}^2$  in the environment, radius  $r_i$  and velocity  $\mathbf{v}_i \in \mathbb{R}^2$ . The SFM is described by a system of linear differential equations

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = \frac{\mathbf{v}_i^0 - \mathbf{v}_i}{\tau_i} + \frac{\mathbf{f}_i + \boldsymbol{\xi}_i}{m_i} \end{cases} \quad (1)$$

$\mathbf{v}_i^0$  is the *driving (desired) velocity* of agent  $i$ , represented by a product of speed  $v_i^0$  and normalised direction  $\mathbf{e}_i^0$ . Usually,  $\mathbf{e}_i^0$  is given by the line joining the current position and the next via point. Importantly, since  $v_i^0$  is by default set to the user’s preferred walking speed,  $\mathbf{v}_i^0$  is time invariant between via points.  $\tau_i$  is the time taken to react to the difference between desired and actual velocity, while  $\boldsymbol{\xi}_i$  is a noise term modelling fluctuations not accounted for by the deterministic part of

the model. The noise term can also serve to avoid deadlocks and hypothesise alternative trajectories. Usually, the  $\boldsymbol{\xi}_i$  is assumed normally distributed. In the absence of the exogenous inputs  $\mathbf{f}_i$  and  $\boldsymbol{\xi}_i$ , the agent’s trajectory simply converges to the driving velocity with time constant  $\tau_i$ .  $\mathbf{f}_i$  is the overall force acting on agent  $i$  resulting from other objects in the environment and is given by

$$\mathbf{f}_i = \sum_{j \neq i} [\mathbf{f}_{ij}^{\text{soc}} + \mathbf{f}_{ij}^{\text{att}} + \mathbf{f}_{ij}^{\text{ph}}] + \sum_b [\mathbf{f}_{ib}^{\text{soc}} + \mathbf{f}_{ib}^{\text{ph}}] \quad (2)$$

The first term on the right-hand side of (2) includes all the forces on agent  $i$  resulting from interactions with other agents:  $\mathbf{f}_{ij}^{\text{soc}}$  is the repulsive social force that inhibits strangers from getting too close,  $\mathbf{f}_{ij}^{\text{att}}$  is the attractive social force that brings agents together,  $\mathbf{f}_{ij}^{\text{ph}}$  is the physical force that exists when two agents come into contact. The second term includes the forces acting on agent  $i$  as a result of fixed environmental obstacles:  $\mathbf{f}_{ib}^{\text{soc}}$  is the social force that inhibits agent  $i$  from getting too close to environmental boundaries,  $\mathbf{f}_{ib}^{\text{ph}}$  is the physical force that exists when agent  $i$  touches the boundary  $b$ .

$\mathbf{f}$  is principally a function of the distance  $d$  between an agent and the other objects in the model.  $d_{ib}$  is the minimum distance between the circumference of agent  $i$  and fixed object  $b$ .  $d_{ij}$  is the distance between the centres of mass of agents  $i$  and  $j$ , i.e., the centres of the discs, while  $r_{ij} = r_i + r_j$  is the “touching distance”. To aid modelling the different force regimes that exist when agents are not in contact and when they touch (i.e. agents  $i$  and  $j$  touch if  $r_{ij} - d_{ij} \leq 0$ ) it is customary to choose the function  $\Theta(r_{ij}, d_{ij}) = \max(0, r_{ij} - d_{ij})$ .

Using these notions, the various repulsive social and physical forces of (2) are defined as follows:

$$\mathbf{f}_{ij}^{\text{soc}} = \{A_i \exp[(r_{ij} - d_{ij})/B_i]\} \mathbf{n}_{ij} \Lambda(\lambda_i, \varphi_{ij}) \quad (3)$$

$$\mathbf{f}_{ij}^{\text{ph}} = k_1 \Theta(r_{ij} - d_{ij}) \mathbf{n}_{ij} + k_2 \Theta(r_{ij} - d_{ij}) \Delta v_{ji}^t \mathbf{t}_{ij} \quad (4)$$

$$\mathbf{f}_{ib}^{\text{soc}} = \{A_i \exp[(r_i - d_{ib})/B_i] + k_1 \Theta(r_i - d_{ib})\} \mathbf{n}_{ib} \quad (5)$$

$$\mathbf{f}_{ib}^{\text{ph}} = -k_2 \Theta(r_i - d_{ib}) (\mathbf{v}_i \cdot \mathbf{t}_{ib}) \mathbf{t}_{ib} \quad (6)$$

$\mathbf{n}_{ij}$  ( $\mathbf{n}_{ib}$ ) is a normalised vector pointing from agent  $j$  (fixed object  $b$ ) to agent  $i$ , i.e., the direction of the repulsive force.  $\mathbf{t}_{ij}$  ( $\mathbf{t}_{ib}$ ) is a normalised vector tangential to the relative movement of agent  $i$  and agent  $j$  (fixed obstacle  $b$ ), i.e., the motion tangential direction.  $\Delta v_{ji}^t = (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{t}_{ij}$  is the tangential velocity difference. The social forces (3) and (5) increase exponentially with reducing distance between objects, with a scale defined by constants  $A_i$  and  $B_i$ . In particular,  $A_i$  is the force acting on agent  $i$  at the touching distance;  $B_i$  is loosely the distance at which the force takes effect. The choice of exponentially increasing forces aligns with the intuition that agents wish to avoid touching (i.e. collisions)

at all costs and that the urgency to achieve this increases dramatically as the agents get closer. Comparison with the behaviour of real pedestrians have borne out this choice for crowd simulations [23,22,21], however we note that the SFM and our approach are not restricted to a particular force profile.

$\Lambda : \mathbb{R}^2 \mapsto [0,1]$  is a function that gives greater weight to the social force (3) arising from the agents in front of (notionally, *seen* by) an agent.  $\lambda_i$  is a parameter that regulates the effect of  $\Lambda$  on agent  $i$ , while  $\varphi_{ij}$  is the angle between the directions  $\mathbf{e}_i^0$  and  $-\mathbf{n}_{ij}$ , i.e., the field of view of the agent. The physical force (4) between agents comprises a repulsive body compression force (first term) that acts in direction  $\mathbf{n}_{ij}$ , plus a frictional force (second term) that acts in direction  $\mathbf{t}_{ij}$  to impede the relative tangential movement of two agents in contact.  $k_1$  and  $k_2$  are constants that define the scale of the physical forces. The physical force (6) between an agent and a fixed object is solely described by a frictional term.

## 2.2 Statistical Model Checking

*Statistical model checking* (SMC [41,25]) is a variety of probabilistic model checking (see e.g. [5]) that avoids an explicit representation or traversal of the state space; estimating the probability of a property from an empirical distribution of executions (simulations) of a system. Given a number of statistically independent simulation traces of a stochastic model, plus an automaton to decide whether a trace satisfies a temporal logic property, it is possible to estimate the probability that the model will satisfy the property. This estimate may be obtained with a specified confidence provided by, e.g., the Okamoto bound [33,25]. It is also possible to efficiently evaluate the truth of an hypothesis without needing to calculate the actual probability using, e.g., the sequential probability ratio test [40,41].

An efficient implementation of SMC is provided by the PLASMA-lab SMC library [35,7], which may be used with arbitrary discrete event stochastic simulators. PLASMA-lab checks properties expressed in bounded linear temporal logic (BLTL), which can express complex behavioural properties with nested temporal causality.

The abstract syntax of a BLTL property is as follows:

$$\phi = \phi \vee \phi \mid \phi \wedge \phi \mid \neg \phi \mid F_{\leq t} \phi \mid G_{\leq t} \phi \mid \phi U_{\leq t} \phi \mid X_{\leq t} \phi \mid \alpha \quad (7)$$

$\vee, \wedge$  and  $\neg$  are the standard logical connectives *or*, *and* and *not*, while  $\alpha$  is an atomic property that is *true* or

*false* in a given state. In practice,  $\alpha$  may be an expression over state variables or externally derived metrics of the trajectories.  $X, F, G$  and  $U$  are temporal operators with a parameter of  $t$  time units (real time or a number of steps).  $X$  is the *next* operator:  $X_{\leq t} \phi$  asserts that  $\phi$  will be true at time (step)  $t$ .  $F$  is the *finally* or *eventually* operator:  $F_{\leq t} \phi$  asserts that  $\phi$  will be true at some time within  $t$  time units.  $G$  is the *globally* or *always* operator:  $G_{\leq t} \phi$  asserts that  $\phi$  will be true at all times within  $t$  time units.  $U$  is the *until* operator:  $\psi U_{\leq t} \phi$  asserts that  $\phi$  will be true within  $t$  time units and  $\psi$  will be true until it is.

It is straightforward to express properties important to ACANTO using BLTL. For example, if  $\phi$  represents collision between the user and any other agent, then  $G_{\leq \infty} \neg \phi$  represents *never* allowing a collision to occur. Similarly, if  $\psi$  encodes that the user is isolated from their group, then  $G_{\leq \infty} (\neg \phi \wedge \neg \psi)$  asserts that the user never collides and is never isolated. Other types of properties can also be encoded. E.g., if  $\phi$  represents reaching the destination and  $\psi$  represents being sufficiently close to a bathroom, then  $F_{\leq t} \phi$  asserts that the user will reach the destination by time  $t$ , while  $\psi U_{\leq t} \phi$  asserts that the user will be close to a bathroom until their destination is reached. BLTL is not equipped with higher order quantifiers, but the behaviour of *all* members of a group can be expressed by a dynamically created explicit conjunction of the individual behavioural specifications. This is possible because it is assumed that the users of the system are known a priori and can be identified.

While BLTL properties are decided deterministically with respect to a particular trace, it is important to note that the result of SMC is an *estimated probability* that the property will hold on an arbitrary execution. In the present context, we use this estimate to choose an immediate action that approximately maximises the probability that the property will be satisfied. Even if the user accepts the suggested action, the system cannot guarantee that the property will *actually* be satisfied. This is in contrast to approaches that synthesise “rigid” motion planners for obedient robots, based on logical constraints (e.g. [26]).

The choice of SMC allows us to use efficient off-the-shelf technology to rigorously define motion planning constraints over continuous time and space, while also allowing for uncertainty and flexibility. Moreover, as a Monte Carlo technique, SMC is able to provide an approximately optimal solution with a level of confidence to suit the available time and resources. As such, we find SMC ideal for our application.

### 3 The ACANTO Project

The ACANTO EU project [1] is developing robotic assistants to improve the mobility and confidence of mobility-impaired and elderly adults. Each robotic assistant (called a *FriWalk*) is a user-propelled “walker” equipped with wheels, sensors, and brakes, that helps to increase the confidence and mobility of individuals by providing both physical and social support. The *FriWalk* is also equipped with a *FriTab*, a tablet computer with a touch-screen interface and a radio link to the ACANTO server infrastructure. The *FriTab* integrates with the *FriWalk* sensors, as well as with the wider ACANTO infrastructure, to provide a variety of support to the user, including: navigational assistance, social networking, social activity planning, therapeutic regime support, and (medical) diagnostic support. The focus in this paper is on the motion planning algorithms required for navigational assistance of groups engaged in therapeutic social activities.

In the remainder, for simplicity, we use the term *walker* to refer to a *FriWalk* with integrated *FriTab*. We use the term *user* to specifically mean a pedestrian using the ACANTO walker, while the term *pedestrian* refers to *all* pedestrians (including users). Where necessary, we use the phrase *unassisted pedestrians* to distinguish those who are not using the ACANTO walker.

The ACANTO system comprises an intelligent environment where various walkers with their sensors operate in coordination with each other and the ACANTO server infrastructure. This infrastructure facilitates collaboration and coordination between walkers, as well as providing information from environmental sensors and access to (temporarily) stored data (e.g. previous sensor observations).

The main sensor on each *FriWalk* is a forward facing RGB-D camera with a minimum effective range of  $\sim 4$  metres. The visual information is processed in the *FriWalk* to recognise agents in the environment as well as determine the location and momentary velocity of all visible agents. Environmental sensors are here assumed to be cameras in the environment that, via the server infrastructure, are also able to detect moving agents and calculate their momentary velocity.

The algorithms considered later in this work focus on how to proceed once sensor information has been curated to yield location and point velocity information for all detected agents. In practice some of this calculation is performed locally on the *FriWalk* (all calculation when the network is not operational).

In the remainder of this section we present the key details of the architecture and algorithm that are common to the navigation support in both the DALi and

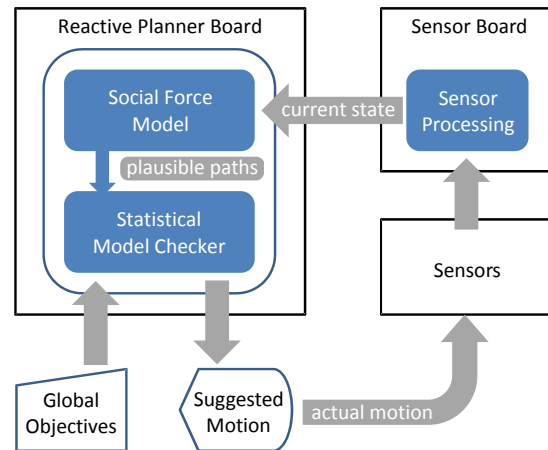


Fig. 1 Reactive Planner architecture.

ACANTO projects. The specific challenges arising from multiple users and the intelligent environment of the ACANTO project are discussed in the next section.

At the core of the navigational assistance engine is a *reactive planner* that operates on a rapid “predictor-corrector” architecture, shown in Figure 1. The reactive planner takes as input the information from sensors and the global objectives associated to the current user and activity (e.g., the global plan plus the user’s constraints and objectives). For simplicity, in the sequel we assume that the global objectives are fixed and given. Sensors provide information that, when processed, locates the user, fixed objects and other pedestrians in the local environment. The reactive planner makes instantaneous navigational suggestions to the user that approximately maximise the probability of satisfying all the global objectives. By re-planning frequently, the reactive planner reduces the impact of measurement noise and model fidelity.

---

#### Algorithm 1: Single User Reactive Planner

---

```

Let  $Move$  be the set of possible navigation suggestions
Let  $Cstr$  be the set of constraints in BLTL
Let  $Self$  be the current position and velocity
while active do
  Let  $Obs$  be the set of current observations
  Set  $action \leftarrow stop$ 
  Set  $prob_{success} \leftarrow 0$ 
  for  $m \in Move$  do
    Set  $model \leftarrow SFM(Obs \cup (Self, m))$ 
    Set  $prob_m \leftarrow SMC(Cstr, model)$ 
    if  $prob_m > prob_{success}$  then
       $action \leftarrow m$ 
       $prob_{success} \leftarrow prob_m$ 
  Display  $action$  to user
  
```

---

Algorithm 1 describes the function of the single user (“selfish”) reactive planner, as used in the DALi project.

The position and (possibly zero) velocity of objects in the environment, denoted  $Obs$ , are identified by the sensors and used to parametrise the human motion model, which in this case is specifically set to  $SFM(Obs \cup (Self, m))$ . The SMC engine simulates multiple future trajectories of the model up to some time horizon, given different hypothesised initial directions  $m$  of the user. Each trajectory is validated against the user’s constraints,  $Cstr$ , and thus the planner is able to estimate the probability of success,  $prob_m$ , for each hypothesised direction  $m$ . The direction suggested to the user, denoted *action*, is that which maximises the probability of success ( $prob_{success}$ ) and minimises deviation from a direct path to the next waypoint.

We suppose that the navigational suggestions given to a user take the form of a compass heading that corresponds to the best hypothesized direction predicted by the SFM and SMC, as defined in Algorithm 1. In the case of the ACANTO group motion planner, the navigational suggestions given to a member of a group are adjusted according to the member’s position within the group. Details of this are given in Section 8.

## 4 Challenges

For a single user with a single walker, the DALi solution has proven successful [11,12]. This holds even though the user is free to ignore or override the navigational advice at any time. The rôle of the walker is merely to *advise* the user. In contrast to many conventional robotic systems, whose robots tend to follow their instructions as best they can, the motion of the DALi and ACANTO robotic assistants is chosen by the users and may be significantly different to the suggested motion.

In ACANTO, the navigational challenges are generalised to social groups that may have many pedestrians, both users and non-users, participating in an activity together. This implies various new challenges, making the single-user approach of the DALi project infeasible. The ACANTO walker must act cohesively and cooperatively in a group, including with unassisted pedestrians and group members who do not necessarily have walkers.

### 4.1 Computation

*Computationally*, a naive extension of Algorithm 1 would make the problem exponential in the number of users. This arises because it would be necessary to consider the combinations of all possible directional choices for all users.

This is further exacerbated by the limitation that the computation must be done rapidly and on the embedded device within the walker, i.e. the FriTab.

### 4.2 Incomplete Information

*Incomplete information* is inherent, since sensors may be occluded, are distributed, and communication cannot be guaranteed. Further, a walker cannot (computationally) model the entire environment, and so must limit the range of sensor information considered.

These limitations in turn will inevitably give rise to situations where different walkers (even in the same group) will have both different and incomplete knowledge.  $Obs$  in Algorithm 1 would contain different values for different walkers in the same group.

### 4.3 Group Cohesion

*Maintaining group cohesion* is non-trivial, since groups of users may be ad-hoc (there is no guarantee that they will respect the intended grouping) and groups of unassisted pedestrians are entirely arbitrary. Indeed, unassisted pedestrians may join groups of users, and vice versa. As a result, identification of members of a group must be computed dynamically. Even given the same information, Algorithm 1 only optimises for *Self*.

Maintaining group cohesion is of particular importance to social groups, where the social goals of global activity plans will fail if group cohesion is lost. However, it is not simple to define what a group is or what group cohesion means, since a large group of pedestrians may split into smaller groups and still maintain group cohesion, despite these subsequent sub-groups being separated.

### 4.4 Frequent Replanning

*Frequent replanning* becomes necessary from the above constraints and the chaotic environment, since otherwise navigation will rapidly be out of date. Even at the relatively low speeds of human motion, frequent updating mitigates the problems created by human motion model inaccuracy, poor sensor performance and communication failure.

This requirement of course adds to the burden of the other challenges. Computation becomes more bound by time and efficiency. Incomplete information can occur more often, since even a short loss of connection may last several planning cycles. Similarly, group cohesion must be effectively stable under rapid replanning cycles

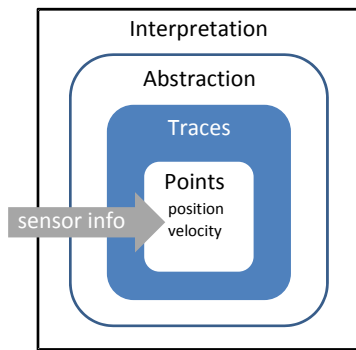


Fig. 2 Group Abstraction Overview.

while also not being damaged by frequent navigational suggestions.

## 5 Solution Overview

This section overviews our approach to addressing the challenges identified in the previous section. Specifically, our solution builds on the single user reactive planner described in Section 3 to address the unique challenges of ACANTO described in Section 4. The key concept of the solution is to abstract away from individuals in favour of groups in the reactive planning algorithm (i.e. replace Algorithm 1), and to exploit additional information from other walkers and sensors in the environment.

Abstracting to groups naturally captures group behaviours and dynamics, while navigation at the level of groups solves the difficulties of attempting to do individual navigation and still have all the individuals in a group maintain group coherency.

For improved behaviour prediction it is necessary to know the intentions of pedestrians, i.e. where they’re trying to go and with whom they’re travelling. This information is well defined for participants of an ACANTO activity, but not for other or unknown pedestrians. Therefore past behaviour is observed to infer future behaviour, as encapsulated by a pedestrian’s *trace*.

Combining the output of sensors on multiple walkers incurs the additional challenge of identifying pedestrians who leave the view of one sensor and appear in another. Pedestrians may also appear and disappear as a result of sensors being obscured, because of communication unreliability, or because users just leave. Thus, the trace inference algorithm, described in Section 6, makes use of a human motion model to reliably link the *behaviour* of pedestrians over much longer times than that between successive (sensor) video frames.

Having inferred a set of traces from multiple observations of users in the environment, it is necessary to infer their de-facto groups. Instantaneous physical prox-

imity is not a sufficient indicator, since two close pedestrians could actually be trying to get away from each other. The group inference algorithm (Algorithm 3) thus uses a notion of proximity that includes both position and velocity: if pedestrians are physically close, walking at the same speed in the same direction, it is reasonable to assume (by definition) that they are walking together. The full technical aspects of how we find an optimal partition of traces into groups are outlined in Section 7. For convenience we make use of *k*-means clustering [28], but note that our approach is not limited to this.

Identifying de-facto groups allows the walker to plan motion at a more efficient level of abstraction. When hypothesising the alternative directions for a number of users of the walker, it is a reasonable compromise to only hypothesise the overall motion of the groups to which users belong. It is not necessary to consider all the possible combinations of suggestions to those within the same group, given that, by virtue of how a group is defined, their motion is strongly correlated. Note that suggestions are nevertheless tailored to the actual position of a user within the group, in order to maintain its “social” structure.

An overview of the group abstraction approach can be seen in Figure 2, the details of which are described in the following three sections. Section 6 details the algorithms used to identify and update traces within the reactive planner. Section 7 presents how to perform group abstraction on the traces to identify social groups from trace behaviour. Section 8 shows how to combine these in an efficient reactive planner that fits within the ACANTO architecture and requirements.

## 6 Trace Inference

The trace inference algorithm (Algorithm 2) constructs sets of active and inactive traces, where a trace is a sequence of time-stamped observations of the position and velocity of pedestrians, detected by the sensors. Active traces are those for which the algorithm has reliably inferred continuity and/or there is currently a pedestrian in the field of view of the sensors (a trace may consist of a single observation). Inactive traces are those for which the algorithm could find no valid continuation, so there is no current view of the corresponding pedestrian. Since inactive traces do not contain a current point, further trace inference applies only to active traces. In practice, a trace may become inactive due to the obfuscation of a sensor or failure of communications. The algorithm assumes that the human motion model makes valid predictions up to a maximum time interval



**Algorithm 2:** Trace inference

---

```

Let  $\mathcal{H}$  be the human motion model
Let  $Act$  be the current set of active traces (initialised
with the first set of observations)
Let  $Inact$  be the current set of inactive traces (initially
empty)
Let  $\Delta t_{max}$  the maximum trace projection time
Let  $D_{trace}$  be the trace matching distance metric
Let  $\theta_{trace}$  be the trace matching threshold distance
while there are new observations do
  Let  $Obs$  be the set of new observations at current
  time  $t$ 
  Let  $Old \subseteq Act$  be the set of traces whose end
  points are older than  $t - \Delta t_{max}$ 
   $Act \leftarrow Act \setminus Old$ 
   $Inact \leftarrow Inact \cup Old$ 
  Let  $Proj$  be the set of points generated by
  projecting the ends of all traces in  $Act$  to time  $t$ 
  using  $\mathcal{H}$ 
  if  $|Obs| < |Proj|$  then
     $\lfloor$  Pad  $Obs$  with dummy entries so  $|Obs| = |Proj|$ 
  else if  $|Proj| < |Obs|$  then
     $\lfloor$  Pad  $Proj$  with dummy entries so  $|Obs| = |Proj|$ 
  Construct a square matrix  $Dist$  of distances
  between  $Obs$  and  $Proj$  using  $D_{trace}$ 
  (set the distance to or from any dummy entry to
  be the maximum non-dummy distance)
  Apply the Hungarian Method to  $Dist$  to find a
  minimum distance set of assignments  $Assign$ 
  Remove from  $Assign$  all assignments involving
  dummy entries
  for  $((trace, projection), observation) \in Assign$  do
    if  $D_{trace}(projection, observation) \leq \theta_{trace}$  then
       $\lfloor$  Append  $observation$  to  $trace$ 
    else
       $\lfloor$  Add  $observation$  to  $Act$ 

```

---

of  $\Delta t_{max}$ , hence traces whose endpoints are older than this become inactive.

The initial set of active traces comprises the initial set of observations, while the initial set of inactive traces is empty. Each step of the algorithm appends new observations to active traces, or starts new active traces with single observations that cannot be assigned to existing active traces. Active traces to which no new point can be assigned become inactive once their endpoint is older than  $\Delta t_{max}$  with respect to the current time. Using the SFM, each iterative step of the algorithm generates a set of projections to the current time from the endpoints of the active traces, then tries to match the projected points to the new observations by finding a minimum distance assignment according to distance metric  $D_{trace}$  (see Section 9.2).

To solve the *assignment problem* of optimally matching new observations to projected points, the implementation makes use of an  $\mathcal{O}(n^3)$  implementation of the Hungarian method [27,8]. First, a square matrix of distances between projected points and the new ob-

servations is constructed, according to distance metric  $D_{trace}$ . The numbers of projected points and active traces may be different, so the smaller of the two sets is padded with dummy entries whose distance to all members of the other set is, by convention, made equal to the maximum observed distance between non-dummy entries. The Hungarian method is guaranteed to find an assignment that minimises the overall distance between the projected points and new observations, but not all individual assignments are close enough to be accepted and some assignments include non-existent (dummy) entries. Observations whose assignment has a distance up to threshold  $\theta_{trace}$  are appended to the ends of the corresponding active traces. Observations whose assignment has a distance greater than  $\theta_{trace}$  become the initial points of new active traces. Assignments involving dummy entries are discarded.

## 7 Group Abstraction

To infer groups, we use  $k$ -means clustering [28] over the set of active traces. The  $k$ -means algorithm partitions a set of  $n \geq k$  data points into  $\leq k$  clusters, according to a problem-specific distance metric over the points. Although the  $k$ -means problem is computationally hard, there are good heuristics that make it expedient for on-the-fly inference application (e.g., the  $k$ -means++ algorithm [4]).

Given a specified value of  $k$ , the  $k$ -means algorithm first defines a set of tentative cluster means (centroids). This may be done randomly or heuristically. It then executes a series of alternating assignment and update steps that (re-)allocate points to clusters, until further steps produce no modifications. Assignment steps assign data points to clusters with the nearest mean, according to the distance metric. Update steps re-calculate the means of the clusters. The  $k$ -means algorithm is guaranteed to terminate, however the results are generally local optima that are dependent on the initialisation. Heuristics therefore focus on finding good initialisations.

Finding an optimal group abstraction is achieved by first defining a group distance metric,  $D_{group}$ , and defining a group cohesion threshold distance,  $\theta_{group} > 0$ , that specifies the maximum permissible distance from the centroid of a group. Note that the group distance metric is more concerned with similar motion than physical proximity, so being close to the centroid implies primarily that members of the group are *moving in the same way*. A priori, the optimal number of clusters (i.e., the optimal value of  $k$ ) is unknown, so the algorithm iterates from  $k = 1$  to  $k = |Act|$ , where  $Act$  is the set of traces to cluster, stopping when the set of clusters

**Algorithm 3:** Group inference

---

```

Let  $Act$  be the set of active traces
Let  $D_{group}$  be the group distance metric
Let  $\theta_{group}$  be the group cohesion threshold distance
Set  $done \leftarrow false$ 
Set  $k \leftarrow 1$ 
while  $\neg done \wedge k \leq |Act|$  do
  Perform  $k$ -means clustering on  $Act$ 
  Let  $Clust$  be the resulting set of  $\leq k$  clusters of traces
   $done \leftarrow true$ 
  for  $cluster \in Clust$  do
    Let  $centroid$  be the centroid of  $cluster$ 
    for  $trace \in cluster$  do
      if  $D_{group}(trace, centroid) > \theta_{group}$  then
         $done \leftarrow false$ 
return  $Clust$ 

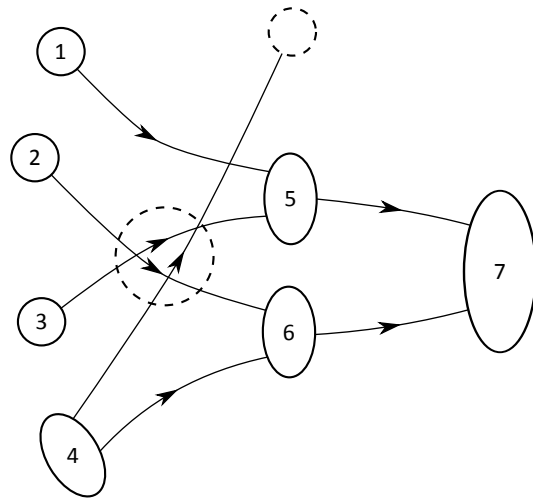
```

---

are sufficiently cohesive. For the purposes of efficient motion planning, minimising  $k$  is desirable, so the algorithm aims to find the fewest number of sufficiently cohesive clusters. To judge the cohesiveness of a cluster, the algorithm calculates the distance between each member and the cluster’s centroid using  $D_{group}$ . If any member of any cluster is too far from its corresponding centroid, the current set of clusters is abandoned and a new set is generated using  $k \leftarrow k + 1$ . The algorithm is guaranteed to terminate because when  $k = |Act|$ , all clusters contain a single element whose distance from the corresponding group centroid is guaranteed to be  $< \theta_{group}$ .

Although the  $k$ -means algorithm partitions data points into Voronoi cells that are disjoint in the multi-dimensional space of the distance metric, groups may physically overlap in the spatial dimensions. This arises, for example, when two groups walking in opposite directions pass through each other. This phenomenon does not occur at the level of individuals and is therefore not considered in the original social force model, however it is nevertheless possible to model it with forces in the SFM framework. For example, it is possible to reduce the repulsive social force between the groups ( $\mathbf{f}^{soc}$  in (2)) and use the physical component ( $\mathbf{f}^{ph}$  in (2)) to model the “friction” between them. To accurately model the momentum of different sized groups, the mass term ( $m$  in (1)) will be the sum of the masses of the individuals. The latency parameter ( $\tau$  in (1)) is also likely to be greater for groups, however some of the latency is included implicitly by the increased mass.

Figure 3 illustrates an hypothetical scenario of groups diverging and coalescing over time, noting that these phenomena are also evident in the automatically-generated visualisations of the output of these on-the-fly algorithms applied to real observations (see Sec-



**Fig. 3** Groups diverging and coalescing.

tion 9). The groups labelled 1, 2 and 3 are assumed to comprise a single pedestrian. Group 4 contains two pedestrians who initially just happen to be walking close to one another in the same direction. Pedestrians 1 and 3 know each other, so they move closer and eventually form group 5. Pedestrian 2 knows one of the members of group 4, so they also move closer to one another and eventually form group 6. The other member of group 4 is just passing through and eventually leaves the view of the sensors (smaller dashed circle). At some time before then, however, he gets very close to pedestrians 2 and 3 (in the region denoted by the larger dashed circle), but no new group is detected because they are all travelling in different directions. The members of groups 5 and 6 are actually part of the same activity, so the system guides them closer, thus eventually forming group 7.

## 8 Group Motion Planning

The trace inference and group abstraction can now be added to the algorithm for the ACANTO reactive planner, as shown in Algorithm 4. Sensor information  $Obs$  is converted into traces using Algorithm 2, and this is used to infer groups  $Grps$  with Algorithm 3. For each walker, i.e., each instance of the planner, the group containing the current user becomes the focus and is considered the only group to which it is possible to make navigational suggestions. The remainder of the algorithm continues as in Algorithm 1, but now yielding the approximately optimal *group* motion, in terms of satisfying the constraints.

Note that the constraints here are assumed to also include group level constraints, such that the algorithm aims to achieve the optimal outcome for both the user and their group. This avoids the selfish optimisation

inherent with the single user reactive planner of Algorithm 1. However, when it is not possible to satisfy all constraints, a balance must be found between prioritising the individual or the group [20].

---

**Algorithm 4: ACANTO Reactive Planner**


---

```

Let Move be the set of possible group navigation
suggestions
Let Cstr be the set of constraints in BLTL
Let Self be the current user trace
while active do
  Let Obs be the set of current observations
  Let Trace be the set of traces inferred by
  Algorithm 2 (using Obs and past state)
  Let Clust be the clusters given by Algorithm 3
  (using Trace  $\cup$  Self)
  Let Grps be weighted groups derived from Obs and
  Clust
  Let Sgrp be the group containing Self
  Grps  $\leftarrow$  Grps  $\setminus$  Sgrp
  Set action  $\leftarrow$  stop
  Set probsuccess  $\leftarrow$  0
  for m  $\in$  Move do
    Set model  $\leftarrow$  SFM(Grps $\cup$ (Sgrp, m))
    Set probm  $\leftarrow$  SMC(Cstr, model)
    if probm > probsuccess then
      action  $\leftarrow$  m
      probsuccess  $\leftarrow$  probm
  Adjust action for user's position
  Display action to user

```

---

Once all the available sensor and server information has passed through trace inference and group abstraction, the resulting groups are used for group motion planning. This is achieved by treating the groups as particles in a SFM, suitably parameterized with appropriate masses ( $m_i$  in (1)) and forces to account for group size, formation, etc. (values in (2)). This group SFM is then passed to the walker's SMC engine, to estimate the optimal suggestion to provide to the user. The SMC engine simulates the consequences of various possible immediate actions that could be taken by the user's group. These constitute an empirical distribution from which it is possible to estimate a group action that jointly maximises the probability of "success" (as defined by the constraints) and minimises deviation from the direct path.

The chosen group action is then tailored to match each individual user's location within the group. Thus, if the individual is on the fringe of a sparse group they can be advised to rejoin, or similarly advised to avoid merging into a dense group. Under other (normal) circumstances, the planner simply tries to maintain a user's position within the group, noting once again that the user is nevertheless free to ignore the advice of the planner.

In general, the advice given to a user is a compass heading that corresponds to the best direction predicted by the SFM and SMC, but tailored according to group cohesion, the user's position within the group and the proposed direction of the group. If the group should move straight ahead and is adequately cohesive, each member will be advised to move straight ahead. If the same group becomes insufficiently cohesive, members will be guided towards the centroid of the group, with their suggested deviation being in proportion to the distance from the centroid. Likewise, if the group becomes too dense, users will be guided away from the centroid, with deviation proportional to the distance from the centroid. The same ideas are applied in a similar fashion when the proposed direction of the group is not straight ahead.

## 9 Experimental Validation

This section presents results of applying our trace inference and group abstraction algorithms (Algorithms 2 & 3) to the ETH Zürich BIWI walking pedestrians dataset. We thus demonstrate their efficiency and efficacy, and show that they are apparently able to infer strictly more information than is contained in the original annotations done by hand.

### 9.1 Dataset

The chosen dataset comprises hand-annotated motion-capture observations of pedestrians in two environments: a hotel lobby and a corridor within the ETH premises. In both environments the observations are made using a fixed camera directly overhead. The annotations link the observations into traces and groups, which the algorithms presented here are able to do automatically. In our experiments the original annotations are used only to compare with the automatically generated annotations, noting here that our algorithms successfully identify *all* the traces and groups identified by hand. Our algorithms work equally well on the two sets of observations, but in this paper we focus on the hotel lobby data, which contains more interesting and varied interactions between pedestrians. In what follows the term 'dataset' refers exclusively to the hotel data.

The dataset contains observations sampled at 2.5fps (0.4s), with observations divided into 27 contiguous intervals separated by more than 0.4s. The excluded times are assumed to be excluded because they contain no *moving* pedestrians, however the results here suggest that some of the omitted frames nevertheless contained *stationary* pedestrians. Moving pedestrians must avoid

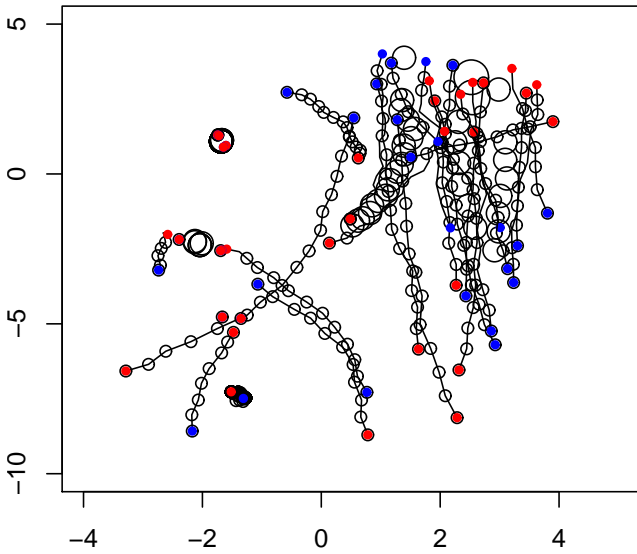


Fig. 4 Group abstraction of interval 34.8s to 82.4s.

stationary pedestrians, so for the purpose of motion planning they cannot be ignored. This is illustrated in the results described below. Note that all times are relative to the first frame of the dataset.

## 9.2 Distance Metrics

The presented results make use of two different distance metrics that are based on both position and velocity. In the case of the trace inference algorithm (Algorithm 2), to decide whether a projected point is “close” to an actual observation, the metric  $D_{trace}$  is based on Euclidean distance in the 4-dimensional space of  $(x, y)$  position and  $discounted(x, y)$  velocity. Precisely, the velocity dimension is divided by 2, thus making position more significant than velocity when inferring traces. The value of 2 was chosen empirically and found to work well, but is not critical. We assume that projected points will have a velocity that is similar to that of the most recent observations, but giving too great an emphasis to velocity-matching risks erroneously cross-linking the trajectories of pedestrians walking in formation.

In the case of the group inference algorithm (Algorithm 3), the metric  $D_{group}$  is based on Euclidean distance in the 4-dimensional space of  $discounted(x, y)$  position and  $(x, y)$  velocity. Hence, in contrast to trace inference, velocity is made more significant than position to infer groups, using  $k$ -means clustering. The premise of our group abstraction motion planning is that pedestrians in a group move in a similar way, so velocity is clearly important. It is also required that pedestrians in a group are spatially proximal, but this is less important. A discount factor of 2 was, once again,

found empirically to be good and not critical. In fact, very similar results were achieved with no discounting, while usable results were obtained by discounting velocity instead of position.

## 9.3 Results

Figures 4, 5, 6 & 7 visualise traces and group abstractions for four intervals from the dataset, produced automatically by our algorithms. The  $x$  and  $y$  axes give the spatial coordinates (in metres) of the groups, with respect to the origin defined in the dataset. The traces inferred by the algorithms are denoted by black lines, marked at their start by blue discs and at their end by red discs. Each black circle denotes an inferred group at a particular time point. A (trivial) group may consist of a single pedestrian (the smallest circles). Most non-trivial groups in these figures consist of pairs of pedestrians (medium-sized circles), with groups containing three pedestrians (largest circle) evident in only Figures 4 & 5. Note that the figures abstract away from time, such that the starts and ends of different traces are not necessarily synchronous, hence intersecting traces in these figures do not necessarily imply a collision.

*Figure 4.* The traces and groups identified by automatic algorithms include all those identified in the hand annotation of the original dataset, as well as other groupings that are useful for motion planning. In particular, the automatic algorithms identify stationary groups near coordinates  $(-1.7, 1.1)$ ,  $(-2.1, -2.3)$  and  $(-1.4, -7.4)$ , where the “group” near  $(-1.4, -7.4)$  is a trivial group comprising a single pedestrian. Moving groups are forced to avoid stationary groups, so the detection of stationary groups is important for prediction. This is illustrated in Figures 5 to 7 and described in their corresponding descriptions, below.

*Figure 5.* First note a stationary group around coordinate  $(1.5, -9)$ . Not apparent from the figure, they first appear at 113.6s and (approximately) maintain their positions for 5.6s, until the end of the interval. Since both their positions and motion are very close, the automatic algorithm correctly identifies these pedestrians as a group. In the original dataset, however, they are not identified as such. Observe that the existence of this stationary group has a significant effect on the motion of the group starting near coordinate  $(2.3, 3.3)$ . For some reason, the moving group heads directly *towards* the stationary group and then splits near coordinate  $(1.7, -4.5)$  to avoid a collision. Importantly, however, up to the point at which the group splits, the group abstraction of the two moving pedestrians provides a good prediction of their behaviour. Following the split,

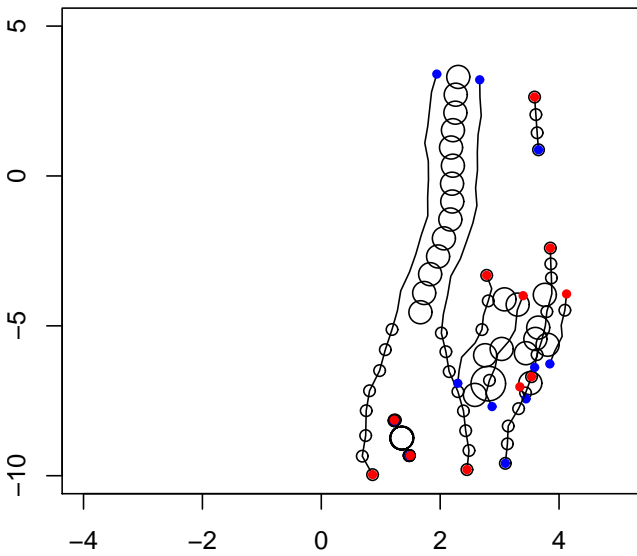


Fig. 5 Group abstraction of interval 110.8s to 119.2s.

the group abstraction *continues* to provide a good prediction of the moving pedestrians’ behaviour because it detects that they are no longer moving together. The other groups in this figure, within the box created by  $x \in [2, 4]$  and  $y \in [-8, -3]$ , are not identified by the hand-annotation of the original dataset, but are nevertheless valid and useful for motion planning.

*Figure 6.* There is once again a stationary group near coordinate  $(2.8, -6.9)$ , which is not identified in the hand annotations. Given that this group also exists in the three intervening intervals between those of Figures 5 and 6, it is presumed that this is the same group.

The motion of the group starting near coordinate  $(2.9, -9.5)$  is interesting in comparison to the long group trajectories in the other figures. In this figure the pedestrians are walking slower (observations made at the same frequency as the other figures make less spatial progress), closer together, and while not following a smooth path, they appear to very closely maintain their separation distance. It may be possible to infer from this that they have a strong social connection. From the perspective of motion planning, they are effectively moving as a single agent, thus fully justifying the group abstraction.

*Figure 7.* The stationary group seen in Figures 5 and 6 seems also to be present in this interval, however the group first appears at time 232.8s, 5.6s after the beginning of the interval, and is not present in the single interval between those illustrated in Figures 6 and 7. From only the recorded observations in the dataset, it is not possible to disambiguate the possibilities that the group moved away and then returned to the same spot, that the camera was temporarily obscured or that there was a data processing error. Temporary occlusions

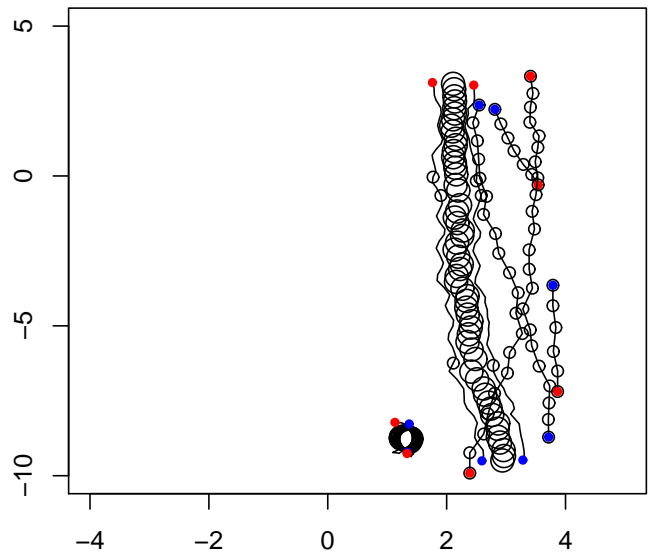


Fig. 6 Group abstraction of interval 177.2s to 200.4s.

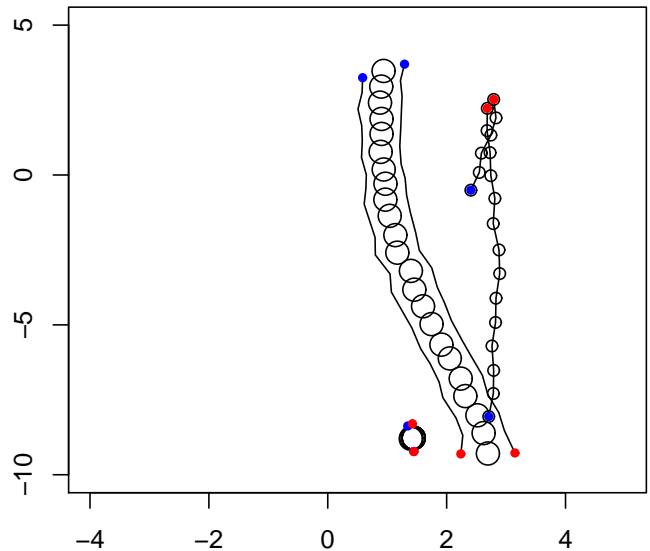


Fig. 7 Group abstraction of interval 227.2s to 236s.

and measurement errors are to be expected, especially when inferring a global view from ground-level cameras mounted on individual walkers, as in ACANTO, and do not invalidate the approach unless the occlusions and errors persist indefinitely.

To conclude this section, we note that our algorithms were able to automatically extract all the information contained in the original manually-generated annotations, in a fraction of a second. We also claim that our algorithms were able to extract additional information that is vital for motion planning.

## 10 Related Work

This section discusses related works and contrasts them with the approaches and results of the current paper.

There are many areas of marginally related works, so we focus on the most relevant: robot assistance and the role of the robot contrasted with classic robotics, and alternate human motion models.

### 10.1 Robot Assistance

One significant difference between the ACANTO project and most areas of robotics is the *assistive*, rather than *prescriptive* nature of its robots. In classical robotics, robots are typically under full control of the motion planner, so it is assumed that they will closely follow the plan or actions deemed most likely to achieve the desired goal [6, 14, 10, 9, 39]. Even in the case of disruptions or problems, robots can be forced to take a particular path, or wait indefinitely until it is “safe” to proceed [10, 9, 39]. Under these circumstances, any actual deviations to the plan tend to be minor and are treated as sensing errors or slippage, etc. This can actually cause a robot to “freeze”, especially in chaotic and busy environments, such as those for which the ACANTO device is intended [38].

In contrast, human users may override navigational advice, make mistakes, or just pursue their own goals, unknown to the navigation engine. Even compared to robotic systems that must consider disruptions [14, 39], the variability of the environment and disruptions envisaged in the ACANTO context may be significantly higher.

Although this scenario is superficially similar to swarm robotics, in the sense of having multiple agents with a common goal, the success criteria are very different. Typically in swarm robotics, success is achieved when some number of robots reach the goal or find a path [13, 29, 32, 31, 42, 30]. However, for ACANTO, an activity is only considered completely successful if *all* the social group members complete the activity, and do so coherently.

### 10.2 Assistance in Navigation

Focusing specifically on collaboration for navigation, there are several existing works that consider similar problems or aspects of the same problem. An overview of many general approaches to shared navigation and control can be found in [36].

In [3] the authors present how to exploit limited robot control to guide a user along a fixed path. This is similar to the work presented here, in allowing the user primary control of a walker, however the need of a prior choice of path and the lack of consideration of

chaotic environments make this approach not immediately applicable to our case.

Somewhat similar work is considered in [38], where the goal is for a robot to act collaboratively with pedestrians in a crowded environment. However, the approach requires significant prior knowledge about the (very small) environment and pedestrians.

In [18] the authors propose a collaboration between a robot wheelchair and a user, where the robot can ask the user for assistance when a task is not feasible for the robot alone. This includes allowing the user to accept or reject plans, and thus provide feedback to the navigational choices made.

In [36] the authors consider how to share the navigational task between a human and a robot. Rather than making either dominant, thus being able to override the other, the action of the robot is a synthesis of the commands chosen by both the human and the robot. We speculate that this approach could be combined with the techniques of [3] to effect shared control of a walker.

One concern highlighted in [20] is the need for trust in the collaboration between robot and human. In particular, in the case that the human and robot disagree on the path to take. In the current paper the solution is to allow the user absolute control, with the robot only providing suggestions. In [3] the approach is to try and force the user, by having the walker prevent “incorrect” directional choices. Although further experiments with actual users would need to be done for a particular domain, recent work [37] has shown that humans do not immediately discard or discount robot assistants that are “faulty”. The results presented in [37] suggest that for minor deviations, the user would be willing to follow the robot’s suggestion.

### 10.3 Human Motion Models

The solution we present here exploits the SFM to model and predict the behaviour of humans; as the model used to extend traces and as the model in SMC. The choice of the SFM is justified by our prior work demonstrating its efficiency and efficacy in this domain [11], and by numerous other applications in the literature. The limitations of the SFM are also well explored in the literature, however we note that our application is not dependent on the SFM. We could, in fact, use any efficient and accurate predictive models of pedestrian or group behaviour.

There is obviously a considerable amount of relevant work in this area. As such, for the sake of brevity, below we confine ourselves to two works that feature interest-

ing modifications and improvements to the SFM, along with two other interesting approaches.

[34] present an improved version of the SFM and discusses some of its inherent limitations. These include strategic decisions (e.g. stopping, in contrast to tactical decisions, e.g. swerving) that need to be made at a higher level.

In [17] the authors consider a refined *headed social force model* (HSFM) that captures more natural human motions in a SFM. This limits motions of particles in the SFM, but yields more human-like trajectories.

In [2] the authors consider an approach based on observing various pedestrians in an environment and “learning” the behaviour of the pedestrians in that environment. Although the results of such prediction could be useful to gain a behavioural profile of a pedestrian, the cost of learning is well beyond the capabilities of the ACANTO walker. It is also unclear whether the approach could be scaled sufficiently to handle all the pedestrians in an environment, and then the behaviour profile communicated to the walkers.

In [19] the authors work on detecting (or creating) groups based on vision (and without annotations). The approach is similar to our work here; exploiting (visual) sensor information, and then clustering based upon location, velocity, and angle of motion. Their approach is to use agglomerative clustering and put constraints on the distances allowed, while our approach is to find the number of groups that do not violate bounds. Their results were able to achieve  $\sim 81\%$  accuracy, while ours achieved  $\sim 100\%$ , albeit on a smaller data set.

## 11 Conclusions

The ACANTO project poses an important challenge for collaborative robotic navigation assistance and human-robot interaction. That is, to provide navigational advice in chaotic environments to groups of mobility-impaired adults engaged in therapeutic activities. Although the navigational challenge for a single user has been previously met, scaling to many users and group activities requires computational efficiency, safety in light of incomplete information, maintaining group cohesion, and rapid replanning.

This paper has described and demonstrated how to provide such navigational advice by inferring behavioural traces of pedestrians in a chaotic environment, abstracting from individuals to groups of pedestrians, representing the groups in the social force model, and then finding a probabilistically optimal navigation solution that maintains group cohesion and safety, using statistical model checking. The technologies presented

here are validated on an open data set of real world observations, showing the effectiveness and efficiency of this solution.

## References

1. ACANTO project web site. <http://www.ict-acanto.eu/>, October 2017.
2. A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F. Li, and S. Savarese. Social LSTM: human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 961–971, 2016.
3. M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli. Path following with authority sharing between humans and passive robotic walkers equipped with low-cost actuators. *IEEE Robotics and Automation Letters*, 2(4):2271–2278, 2017.
4. D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
5. C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
6. M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and autonomous systems*, 41(2):89–99, 2002.
7. B. Boyer, K. Corre, A. Legay, and S. Sedwards. PLASMA-lab: A Flexible, Distributable Statistical Model Checking Library. In *Proceedings of QEST*, volume 8054 of *LNCIS*, pages 160–164. Springer, 2013.
8. R. Burkard, M. Dell’Amico, and S. Martello. *Assignment Problems, Revised Reprint*. Other titles in applied mathematics. Society for Industrial and Applied Mathematics (SIAM), 3600 Market Street, Floor 6, Philadelphia, PA 19104, 2009.
9. M. Čáp, P. Novák, A. Kleiner, and M. Selecký. Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering*, 12(3):835–849, 2015.
10. M. Čáp, P. Novák, M. Selecký, J. Faigl, and J. Vokffnek. Asynchronous decentralized prioritized planning for coordination in multi-robot system. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3822–3829. IEEE, 2013.
11. A. Colombo, D. Fontanelli, A. Legay, L. Palopoli, and S. Sedwards. Motion planning in crowds using statistical model checking to enhance the social force model. In *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013, December 10-13, 2013, Firenze, Italy*, pages 3602–3608. IEEE, 2013.
12. DALi project web site. <http://www.ict-dali.eu/dali>, September 2017.
13. F. David. Map-based navigation in mobile robots: I. A review of localization strategies: *Cognitive Systems Research*, 4:243–282, 2003.
14. V. R. Desaraju and J. P. How. Decentralized path planning for multi-agent teams with complex constraints. *Autonomous Robots*, 32(4):385–403, 2012.
15. ETH Zürich BIWI walking pedestrians open dataset web site. <http://www.vision.ee.ethz.ch/datasets/>, October 2017.

16. F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo. When Helbing Meets Laumond: The Headed Social Force Model. In *Proc. IEEE Int. Conf. on Decision and Control (CDC)*, pages 3548–3553, Las Vegas, Nevada, US, Dec. 2016. IEEE.
17. F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo. Walking Ahead: The Headed Social Force Model. *PLOS ONE*, 12(1):1–23, 01 2017.
18. C. Galindo, J. González, and J.-A. Fernández-Madrigal. Control architecture for human-robot integration: application to a robotic wheelchair. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 36 5:1053–67, 2006.
19. W. Ge, R. T. Collins, and B. Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):1003–1016, 2012.
20. T. Given-Wilson, A. Legay, and S. Sedwards. Information security, privacy, and trust in social robotic assistants for older adults. In T. Tryfonas, editor, *Human Aspects of Information Security, Privacy and Trust - 5th International Conference, HAS 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings*, volume 10292 of *Lecture Notes in Computer Science*, pages 90–109. Springer, 2017.
21. D. Helbing, I. Farkas, P. Molnár, and T. Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. In M. Schreckenberg and S. D. Sharma, editors, *Pedestrian and Evacuation Dynamics*. Springer, 2002.
22. D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, September 2000.
23. D. Helbing, I. J. Farkas, and T. Vicsek. Freezing by heating in a driven mesoscopic system. *Phys. Rev. Lett.*, 84:1240–1243, 2000.
24. D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.
25. T. Héroult, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *Verification, Model Checking, and Abstract Interpretation (VMCAI 2004)*, volume 2937 of *LNCS*, pages 73–84. Springer, 2004.
26. H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, Dec 2009.
27. H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
28. J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.
29. M. Mamei and F. Zambonelli. Physical deployment of digital pheromones through rfid technology. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 281–288. IEEE, 2005.
30. H. Mo and L. Xu. Research of biogeography particle swarm optimization for robot path planning. *Neurocomputing*, 148:91–99, 2015.
31. S. Nouyan, A. Campo, and M. Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1):1–23, 2008.
32. K. J. O’Hara and T. R. Balch. Pervasive sensor-less networks for cooperative multi-robot tasks. In *Distributed Autonomous Robotic Systems 6*, pages 305–314. Springer, 2007.
33. M. Okamoto. Some inequalities relating to the partial sum of binomial probabilities. *Annals of the Institute of Statistical Mathematics*, 10:29–35, 1959.
34. D. R. Parisi, M. Gilman, and H. Moldovan. A modification of the social force model can reproduce experimental data of pedestrian flows in normal conditions. *Physica A: Statistical Mechanics and its Applications*, 388(17):3600 – 3608, 2009.
35. PLASMA-lab SMC library web site. <http://project.inria.fr/plasma-lab>, October 2017.
36. A. Poncela, C. Urdiales, E. J. Pérez, and F. S. Hernández. A new efficiency-weighted strategy for continuous human/robot cooperation in navigation. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 39(3):486–500, 2009.
37. M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn. Would you trust a (faulty) robot?: Effects of error, task type and personality on human-robot cooperation and trust. In J. A. Adams, W. D. Smart, B. Mutlu, and L. Takayama, editors, *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI 2015, Portland, OR, USA, March 2-5, 2015*, pages 141–148. ACM, 2015.
38. P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *I. J. Robotics Res.*, 34(3):335–356, 2015.
39. M. Vavrinec and M. Schaefer. Prioritized planning for road vehicles coordination. 2015.
40. A. Wald. Sequential Tests of Statistical Hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.
41. H. Younes and R. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *CAV*, volume 2404, pages 23–39. Springer, 2002.
42. Y. Zhang, D.-w. Gong, and J.-h. Zhang. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*, 103:172–185, 2013.