Health, demographic change and wellbeing

Personalising health and care: Advancing active and healthy ageing

H2020-PHC-19-2014

Research and Innovation Action

ACANTO

A CyberphysicAl social NeTwOrk using robot friends

Deliverable 5.1.1

User reactive planning

Deliverable due date: 11.2015	**Actual submission date: day.month.year**
Start date of project: February 1, 2015	**Duration: 36 months**
Lead beneficiary for this deliverable: INRIA	**Revision: 0.9**
Authors: Sean Sedwards, Thomas Given-Wilson, Axel Legay	
Internal reviewer: Daniele Fontanelli	

The research leading to these results has received funding from the European Union's H2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement n° 643644.

Dissemination Level		
PU	Public	
CO	Confidential, only for members of the consortium (including the Commission Services)	**X**

Contents

Executive Summary

This deliverable concerns behaviour classification algorithms for use by state of the art dynamic planning. We develop efficient, autonomous online algorithms that infer in real time the parameters of human motion models developed by work package 2. We present an hierarchical framework that analyses the local environment and classifies the behaviour of moving agents or groups of agents. The framework currently takes as input a series of snapshots of instantaneous behaviour observed by the sensors (position and velocity of pedestrians), constructing traces that evolve over time. The behaviour evident in the traces is clustered to infer groups, which reduces the computational complexity of the motion planning problem. The savings arise because the motion of pedestrians within a group is strongly correlated. The level of abstraction may be selected according to available resources, such that a group may consist of a single pedestrian. The framework is therefore able to recognise behavioural templates at the level of individuals and groups. We also consider new templates that are specific to groups (e.g., "passing through"), which we propose to identify and quantify using efficient graph-theoretic metrics. Improved sensor technology may in future allow the framework to do less work, however it will nevertheless continue to "fill in the gaps" of sensory limitations. Finally, we identify concrete directions of ongoing research.

Framework for behaviour classification

1 Introduction

Work package 5 (WP5) provides the execution support for social activities and includes the development of algorithms and a logical engine to facilitate the motion planning necessary for group activities. In general, the planned motion will aim to keep a group together (to allow direct social interaction) and to ensure the therapeutic objectives of the planned activity are achieved efficiently and with minimal discomfort to the users. The tasks are divided into three strands: activity planning (Task 5.1), reactive planning (Task 5.2) and activity monitoring (Task 5.3). The goal of this deliverable is to provide algorithms that enhance the predictive power of the reactive planner by classifying complex behavioural templates inferred from the prior trajectories of observed pedestrians. The information is used to parametrise human motion models to more accurately predict future behaviour.

Activity planning generates the a priori global plan of therapeutic and social activities defined by the activity generator. Reactive planning refers to local motion planning that copes with the actual conditions encountered by the users, given an a priori activity plan. In addition to accounting for unforeseen changes to the environment and other pedestrians who are not part of the activity, the reactive planner also accounts for the random, potentially uncooperative behaviour of users of the system. Activity monitoring is performed in real time and ensures that the concrete suggestions offered to the users will achieve the goals of the activity with high probability.

The specific requirement of the deliverable is to develop online algorithms that infer in real time the parameters of human motion models developed by work package 2 (see deliverable D2.1.1). Although the ACANTO system will have a powerful centralised infrastructure, communication latency and potential interruption require that reactive motion planning is both autonomous and cooperative. The algorithms must therefore be efficient because the motion planning problem is complex and the algorithms will be executed on low powered embedded hardware. In general, we require the system to be robust and able to take advantage of increased computational power and additional information as these become available.

In what follows we present the basis of an hierarchical framework to analyses the local environment and classify the behaviour of moving agents or groups of agents. The framework takes as input a series of instantaneous snapshots of behaviour observed by the sensors. From these it constructs traces that evolve over time. In future, following developments of the ACANTO sensor technology, the framework will take traces or partial traces as input.

The more complex behaviour evident in the traces is clustered to infer grouping and other metrics that also evolve over time. The interpretation of these dynamic metrics allows ever more complex patterns of behaviour to be classified. To improve efficiency, we propose a group-based model abstraction that takes advantage of the fact that the motion of people walking together is strongly correlated. We thus motion plan at the level of groups, while incorporating a sliding level of abstraction that allows groups to consist of a single pedestrian.

1.1 Reactive planning

Figure 1 gives a diagrammatic overview of the ACANTO reactive planner, the key elements of which are summarised below.
The global objectives, comprising the specification of the chosen activity and the preferences of the users, is provided as input a priori. During the course of the ensuing activity, sensors locate the users and other pedestrians with respect to the fixed objects in the environment. This information is used to parametrise a predictive stochastic model of human motion based on the social force model (SFM [15, 4]). The SFM is summarised in Section 2 and described in detail in Section 3. This model is used to simulate multiple future trajectories with respect to alternative immediate behaviour of the users. The sets of simulated trajectories corresponding to each alternative immediate behaviour are validated against the global objectives using statistical model checking (SMC). The immediate behaviour that maximises the probability of achieving the global objectives is recommended to the users.

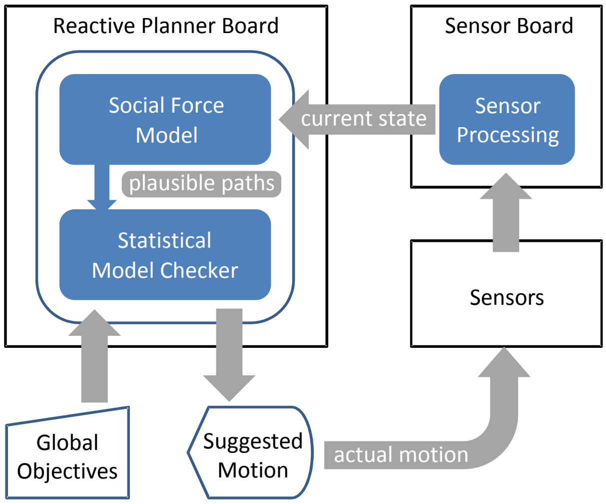

Figure 1: Overview of reactive planning.

The measurements of the sensors contain an element of noise, but for the purposes of the SFM are treated as deterministic best approximations. To account for their potential inaccuracy and the fact that the model is necessarily an incomplete representation of reality, we include a random noise term (ξ in (1), see Section 3) that allows the SFM to explore non-smooth behaviour and behaviour that is not explicitly modelled by forces. Simulations of the model are therefore samples of a random variable and it is for this reason that we then use SMC to estimate the probability of "successful" trajectories.
Our reactive planner is in fact a combination of reactive and predictive planning. A purely reactive approach might be adequate if we could guarantee perfect sensing with no latency. On the other hand, with a perfect predictive model we would have minimal need to sense the environment. Since neither of these are feasible, we adopt a "predictor-corrector" approach. We make a recommendation to the user based on a prediction with an efficient human motion model (i.e., the SFM), then correct our recommendation with updated predictions as the user progresses. Using this approach we significantly improve on the performance of the SFM [4] and can accommodate unpredictable eventualities that would be difficult to include in any reasonable model.
In [5] 'behavioural templates' were identified that characterise modes of pedestrian interaction that are not well modelled by "smooth flows" of individuals moving at constant speed (deliverable D2.1.1 identifies other interesting behaviour). For example, a pedestrian slowing or stopping to give way to a faster pedestrian is common in crowded situations, while repeatedly stopping and starting is common in queues. The use of random behaviour to model these templates cannot accurately predict the true causality, so the performance of the reactive

planner is reduced. We anticipate that this reduction will be critical in the demanding group motion planning required by ACANTO. Note that a mechanism for implementing stopping and starting has been investigated in the case study of deliverable D2.1.1. These case studies are also based on stochastically interacting multiple versions of the SFMs (supporting complex user and group behaviours), thus the solution proposed here is a valid candidate for the future results of this work package.

1.2 Group motion planning

The notion of groups of pedestrians and their interaction is key to ACANTO. While the reactive planning approach described in Section 1.1 has been shown to be efficient on embedded hardware for the case of a single user [4], using the same ideas with a group of users leads to a potential exponential explosion of hypothesised initial behaviour. Given that each user may go left, right or straight with respect to their current direction, there are a minimum of $3^{\#\text{users}}$ alternatives to try. In practice there are also different degrees of left and right choices, so the number of alternatives is much higher. Fine tuning the computational efficiency of the reactive planning algorithms will be the subject of future work, however our proposed approach is to construct the group behaviour compositionally and to first hypothesise alternative initial behaviour of groups as a whole. Suggestions to individuals within the group will aim to respect the group motion and maintain the group cohesion. Similar ideas for crowd simulation have been explored in [23].
One of the principal concerns of this deliverable is therefore the detection of groups. Although the group of users involved in the activity may be known a priori, this notional grouping may not adequately reflect the actual grouping for the purposes of motion planning. For example, it is known that when part of a large group people often prefer to walk together in smaller groups of between two and four to facilitate conversation [20]. In addition, for technical reasons (e.g., temporary loss of network connection), it may not always be possible for an individual FriWalk to recognise all of the other members of a group. Moreover, most people in the environment are likely not part of the the therapeutic activity, but may nevertheless be moving in ad hoc groups [20]. It is therefore necessary to infer grouping directly from the trajectories of pedestrians.

2 Related work

This section considers some related work relevant to the technology adopted by this deliverable, without attempting to be exhaustive.

2.1 Social Force Model

We employ the social force model (SFM) of [15, 14, 13, 12] at various levels of our reactive planner, to predict the behaviour of pedestrians in crowds. Our principal reasons for choosing the SFM are its proven success, its flexibility and its efficiency. The concept of forces representing psychological desires is implicit in many human motion models, but is made explicit in the SFM. The SFM recognises that pedestrians are constrained by the physical laws of motion and also by social ‘laws’ that can also be modelled as physical forces. The original successful application of the SFM was to panic situations [13]. Since then the SFM has had other successful applications, including our own [5, 4], where the assumptions of panic are not necessarily valid or useful. See deliverable D2.1.1 for a survey of related models and a full technical description. A brief mathematical description of the SFM is given in Section 3.

2.2 Clustering Trajectories

There are many works that consider clustering trajectories in the domain of motion planning and prediction, although the constraints of our scenario and our “predictor-corrector” approach make most of them significantly

different.
Initial approaches such as [10] exploit k-means clustering [18] as a simple way to cluster trajectories together. They use the same key concepts that we propose to successfully cluster and thus detect motions represented as trajectories.
Of course there are many ways to consider trajectory clustering, [19] survey several competing approaches to clustering trajectories on both vehicular and pedestrian traffic. They conclude that many different techniques (including k-means) are all appropriate, with none being significantly better than others. Thus since k-means is widely understood and used in this area, we consider it reasonable under this light to exploit in our solution.
The authors of [22] consider real time trajectory clustering as an approach to probabilistically predict behaviour. Although our approach does not attempt to fully predict behaviour to the same scale, we face similar constraints operating in real time and with incomplete trajectory information. Similar to our work, the authors use vectors and predict the next point in real time. They then extend trajectories by using Euclidean distance to find the right point. The authors then try to predict future behaviour by clustering the current trajectory into known classes. Lessons from this last part may be applicable to the behavioural template predictions and planning applied to groups in our proposed approach.
In [17] the authors consider sub-trajectories and clustering upon sub-trajectories. This supports clustering of partial trajectories in a manner than is useful for agents that do not have a single behaviour or classification. This is particularly pertinent to understanding human motion models within known spaces since we can classify common paths, behaviours, and groups that may not hold for the entirety of a trajectory. In a sense this is close to our larger goal of having dynamic group membership, since we can consider a cluster sub-trajectory to be indicative of group membership.
More recently [2, 27] similar approaches have been used to detect trajectories for route planning over a known space. In [2] this is used to learn paths for vehicular traffic (in particular taxis and traffic jams) using clustering to abstract away small details of the route and focus on larger scale planning and prediction. This was exploited to design routing paths that improved the efficiency of the network.
In [27] the authors consider the problem or predicting the location (and possible interception) via trajectories. They simplify various points into trajectories and then cluster these, allowing for simplification of the overall motion, and then easier prediction and classification of the path of an agent. This allows for far more accurate path prediction of agents, and the authors show they can vastly improve interception. Although most of our planning is on recognising paths and groups, the lessons learned from planning interceptions could be exploited to create or regain groups.
Similar approaches have been used in other route domains such as air traffic [11]. Here trajectories are clustered and used to infer information about the status of the network and the availability of routes. This is similar to our proposed approach in the sense that we can consider trajectories of agents to be indicative of behaviour, specially when moving through the larger scale routes of the activity planner. Also in both domains we consider collisions to be undesirable.

2.3 Group Behaviours

The understanding and modeling of groups has two main aspects in relation to this deliverable. By understanding how groups form and what constitutes a group the algorithms can be made to more accurately detect and cluster agents into groups. Also, once groups are detected, behaviour of the group and within the group can be exploited to improve prediction in the reactive planner, and augment the SFM and SMC to yield more accurate guidance to the FriWalk user.
It is widely recognised that the majority of of people travel in groups [6, 20, 7, 9, 1]. Indeed, most groups are either two person dyads (around 68% of groups) or three person triads (another 22% of groups) with the rest (approximately 10%) being in groups of four or more persons. Studies have also shown that larger groups consisting of more than three tend to split into separate smaller groups [16, 6]. It has also been observed

that there can be sub-groups within a larger group [7, 6]. Although interesting, this observation is likely to be handled automatically in our approach by fine tuning the abstraction parameters, selecting for either large groups or separate small groups.
The formation of groups from individual agents can occur in a number of ways. In [8] crowds are formed into groups by common goals, an approach which can be applied here for FriWalk users who have similar goals known to the activity planner. Another approach is to consider proximity [3] as an indicator of group formation. This corresponds to our proposed approach in the reactive planner and the clustering by similar trajectories.
Once a group is formed (and assumed absent of sub-groups [7, 6]) the formation of the group members is of significant interest. This applies to both the shape of the group to be represented in the SFM and used by the reactive planner, and to the motion of the groups' members. In general, dyads move abreast [20, 7, 1, 6], with triads forming either a V or U-shaped formation [20, 7, 1]. Larger groups tend to split into sub-groups [16, 6] that in turn can be considered abreast or V/U-shaped.
The movement of groups has been studied before in [20, 24, 25, 26]. Although [24] assumes that all groups move roughly abreast, more recent works consider the possible deformation of the group members to avoid collisions [25, 26]. In [25], studies of group movement indicated that groups tend to form a hinged line and maintain proximity by hinging to deform around obstacles and avoid collisions. Further in [26] it was recognised that the larger representation of a group does not need to entirely avoid a collision, as members of a group may move within the overall group footprint to allow another individual or group to traverse the group's footprint without colliding with any group member.
Although not part of the current deliverable, we may in future make use of these ideas to better model and preserve the internal structure of groups.

3 Social force model

To facilitate the discussion of technical details, we briefly review the mathematics of the social force model [15, 14, 13, 12]. See also deliverable D2.1.1.
The social force model (SFM) combines real and psychological forces to predict the behaviour of pedestrians in crowds, under normal and panic situations. The model recognises that pedestrians are constrained by the physical laws of motion and also by social 'laws' that can be modelled by external forces. The model considers an environment comprising fixed objects (walls) and moving agents (pedestrians) that respond to attractive and repulsive forces.
The model is constructed in two dimensions, with agents represented as circular discs. In what follows we adopt the convention of denoting vectors in bold type. Thus, agent i has mass m_i centred 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 model for the i-th agent is given by

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_i \\ \dfrac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \dfrac{\mathbf{v}_i^0 - \mathbf{v}_i}{\tau_i} + \dfrac{\mathbf{f}_i + \boldsymbol{\xi}_i}{m_i} \end{cases} \tag{1}$$

$\mathbf{v}_i^0$ is the *desired velocity* of agent i, represented by a product of speed amplitude v_i^0 and normalised direction $\mathbf{e}_i^0$. In general, both elements of the driving velocity are functions of time, as explored in the case study of deliverable D2.1.1. The latency parameter τ_i models the time taken to react to the difference between desired and actual velocity, while $\boldsymbol{\xi}_i$ is a *noise term* (a random variable) that models random fluctuations not accounted for by the deterministic part of the model. The inclusion of the noise term makes the model stochastic, such that a different trajectory is generated each time (1) is solved. This allows the application of SMC and serves two important purposes. Firstly, it avoids deadlocks that might arise if, by chance, some of the deterministic forces are equal and opposite. Secondly, since the predicted behaviour may be critically dependent on the initial state,

which may in turn contain significant measurement error, the noise term allows the system to consider a much wider range of plausible future trajectories.
$\mathbf{f}_i$ is the force acting on agent i resulting from other objects in the environment and is given by

$$\mathbf{f}_i = \sum_{j \neq i} [\mathbf{f}_{ij}^{\mathrm{soc}} + \mathbf{f}_{ij}^{\mathrm{att}} + \mathbf{f}_{ij}^{\mathrm{ph}}] + \sum_{b} [\mathbf{f}_{ib}^{\mathrm{soc}} + \mathbf{f}_{ib}^{\mathrm{ph}}] + \sum_{c} \mathbf{f}_{ic}^{\mathrm{att}}. \tag{2}$$

The first term on the right-hand side of (2) includes all the forces on agent i resulting from interactions with other agents: $\mathbf{f}_{ij}^{\mathrm{soc}}$ is the repulsive social force that inhibits agents getting too close, $\mathbf{f}_{ij}^{\mathrm{att}}$ is the attractive social force that brings friends together, $\mathbf{f}_{ij}^{\mathrm{ph}}$ is the physical force that exists when two agents touch. The second summation includes the forces acting on agent i as a result of the boundaries of fixed objects in the environment: $\mathbf{f}_{ib}^{\mathrm{soc}}$ is the social force that inhibits agent i from getting too close to boundaries, $\mathbf{f}_{ib}^{\mathrm{ph}}$ is the physical force that exists when agent i touches boundary b. Finally, $\mathbf{f}_{ic}^{\mathrm{att}}$ is the attractive social force that draws agent i towards fixed objects of incidental interest (shops, cafés, toilets, etc.).
In general, the force acting on any agent is calculated with respect to the distance between its centre of mass and all other visible objects. In the case of social forces, visible refers to the forward-biased field of view of the pedestrians. In the case of physical forces, visible refers to the omnidirectional field of view of the sensors. Since the model mixes both notional (social) and real forces, the mass m_i is notionally the real mass of agent i. Other parameters can be used to model the unique characteristics of individual agents. For example, the latency factor τ_i can be used to model the possibly reduced mobility of agent i. Full details of these and other parameters can be found in [13, 12]. In [4] we show how the model may be parametrised from captured motion.

4 Our approach

The ACANTO reactive planner draws on the architecture and technology of the 'short term planner' developed in the DALi project. In DALi the short term planner is user-centric and essentially selfish. Visual sensors (based on the Kinect system) attached to the user's walker locate moving agents (pedestrians) in the local environment. Using the architecture shown in Fig. 1, the short term planner then suggests a direction that maximises the probability of satisfying the users objectives, but without considering the objectives of other pedestrians, who might also be using the DALi platform.
In ACANTO the reactive planner will be collaborative and cooperative. Sensor information obtained by each user will be shared between other users of the ACANTO platform, effectively giving the planner a much wider view. The planner will explicitly consider group motion and identify pedestrians who are part of the same social activity as the user. Due to potential communication latency or interruption, planning will nevertheless be local to the user. The significantly increased complexity of the planning task thus necessitates an approach that is efficient.
Figure 2 shows a diagrammatic representation of our behaviour classifier.
The classifier takes as input sensor information provided by the sensor board, as shown in Figure 1. This information will at least contain the estimated positions and velocities at a given time point of moving agents in the vicinity of the sensors of a number of users of the ACANTO platform. The current sensor technology (based on Kinect) does not recognise the identity of individuals between consecutive readings by the same sensor. Future developments may allow the sensor technology to directly infer traces or partial traces and to identify users of the FriWalk, reducing the computation from sensor input to traces and clusters.
Combining the output of multiple sensors 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. Our trace inference algorithm, described in Section 4.1, therefore makes minimal a priori assumptions about the data, but will take

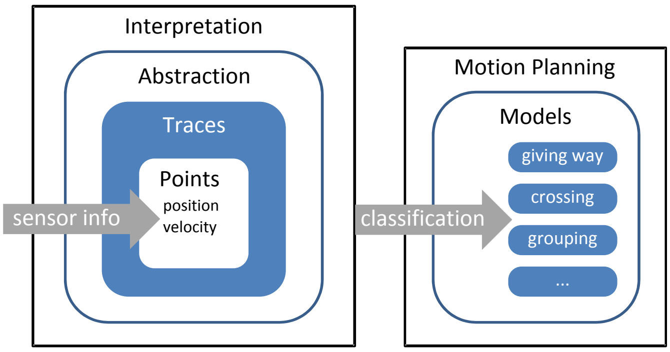

Figure 2: Hierarchical structure of behaviour classifier.

advantage of whatever information is available. If no additional input is available from other users, an individual reactive planner can still function using the information provided by its local sensors, along the lines of the DALi short term planner.

Having inferred a set of active traces, the classifier then clusters them into groups of traces with characteristics that imply the pedestrians are or will be moving as a group. Mere proximity is not a sufficient indicator, since two close pedestrians may be trying to get away from each other. The classifier therefore also considers velocity and acceleration (inferred from successive observations of velocity or predicted by the SFM). It is possible for the framework to include higher level information (e.g., we may know that two pedestrians are part of the same activity group), but if pedestrians are close and moving in a similar direction at a similar speed, for the purposes of motion planning they are already moving as a group, regardless of whether they are involved in the same activity. The technical aspects of this clustering are outlined in Section 4.2, where we also consider the possibility of groups that are not spatially disjoint.

Identifying de facto groups allows us to plan motion at a more efficient level of abstraction. When hypothesising the alternative directions for a number of users of the platform, we believe that it is a reasonable compromise to only hypothesise the overall motion of the groups to which they belong. We feel it is not necessary to consider all the possible combinations of suggestions to those within the same group given that, by virtue of how we define a group, their motion is strongly correlated. Note that suggestions are nevertheless tailored to the actual position of an individual within the group, in order to maintain its "social" structure. A further advantage of this approach is that we may also identify behavioural templates at the level of groups, rather than at the computationally prohibitive level of individuals. We may also quantify temporal properties over traces of group-related metrics. These concepts are described in Section 4.3.

Finally, it is important to note that a group may comprise a single pedestrian, so our framework allows us to choose a level of abstraction that is appropriate for the available computational power. In general, our approach is to plan the motion of an individual against an abstraction of the environment that may be as detailed or complex as the available computational capacity allows.

4.1 Trace generation

The trace generation algorithm constructs sets of active and inactive traces, where a trace is a set of points and a point is the timestamped position and velocity of a pedestrian observed 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 point). 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 classification only applies to active traces. In practice a trace may become inactive due to the obfuscation of a sensor or failure of communications. Our classification is biased towards recent observations, such that knowing the previous history is not necessarily useful. At this stage of development we therefore choose to start new traces rather than make long, potentially unreliable projections from inactive traces.

The initial sets of active and inactive traces are empty. Each step of the algorithm appends new points (observed by the sensors at the current time) to active traces or starts new active traces with points that cannot be assigned to existing active traces. Active traces to which no new point can be assigned become inactive. The initial iterative step of the algorithm therefore assigns all the observed points to the set of active traces. Subsequent iterative steps construct a set of points that are the end points of the active traces projected to the current time. We have implemented both linear projection (very efficient and accurate for short time steps) and projection using the SFM (more accurate for longer time steps). The algorithm then matches the projected points to the newly observed points by minimising the distance (as given by function Dist in Algorithm 1) between them, where the distance may simply be the Euclidean distance in two spatial dimensions or a more complex notion of distance involving velocity (which contains both amplitude and direction) and other relevant information.

In general, the matching of the set of new points to the set of projected points is a clustering problem that results in three disjoint sets of points. S_1 is a subset of new points that cannot be matched to active traces. S_2 is a subset of new points matched to the projected points of a subset of active traces. S_3 is a subset of projected points that cannot be matched to new points. The elements of S_1 become the initial points of new active traces. The elements of S_2 are appended to the active traces to which they have been matched. The active traces that correspond to the elements of S_3 are removed from the set of active traces and added to the set of inactive traces.

Measurement errors in the observation points, in projection and aggregation of points, and an imperfect predictive model may result in ambiguity, i.e., more than one projected point could be the predecessor of a new point or vice-versa. We resolve this by ensuring that the elements of the three sets have the following properties. The elements of S_1 are either too far from any projected point (i.e., have a distance greater than some pre-defined threshold) or the projected points to which they are adequately close are closer to another new point. The elements of S_2 are the closest points to their corresponding projected points. The elements of S_3 are either too far from any new point (distance greater than pre-defined threshold) or the new points to which they are adequately close are closer to other projected points (i.e., they are in S_2). Including velocity in the notion of distance significantly reduces the occurrence of ambiguity.

Algorithm 1 summarises our trace generation approach. To simplify the presentation the algorithm makes use of three functions. Function $\mathrm{Dist}(p_1, p_2)$ gives the distance between points p_1 and p_2. Function $\mathrm{Match}(p)$ gives the trace to which point p has been matched. Function $\mathrm{Trace}(p)$ gives the trace which contains point p.

Algorithm 1: Trace generation

Let Act be the active set of traces, initially empty
Let $Inact$ be the set of inactive traces, initially empty
while *there are new points* **do**
    Let New be the set of new points at current time t
    Let $Proj$ be the set of end points generated by projecting all traces in Act to t
    Compare New with $Proj$ to produce three sets of points:
        $S_1 = \{n \in New : \forall p \in Proj, \mathrm{Dist}(n,p) > \epsilon \vee \exists n' \neq n : \mathrm{Dist}(n',p) \leq \mathrm{Dist}(n,p)\}$
        $S_2 = \{n \in New : \forall p \in Proj, \mathrm{Dist}(n,p) \leq \epsilon \wedge \nexists n' \neq n : \mathrm{Dist}(n',p) < \mathrm{Dist}(n,p)\}$
        $S_3 = \{p \in Proj : \forall n \in New, \mathrm{Dist}(n,p) > \epsilon \vee n \in S_2\}$
    $Act \leftarrow Act \cup S_1$
    $\forall n \in S_2, \mathrm{Match}(n) \leftarrow \mathrm{Match}(n) \cup \{n\}$
    $\forall p \in S_3, Act \leftarrow Act \setminus \{\mathrm{Trace}(p)\}$
    $\forall p \in S_3, Inact \leftarrow Inact \cup \{\mathrm{Trace}(p)\}$

4.2 Group abstraction

To infer groups we use k-means clustering [18] over the set of active traces. The k-means algorithm partitions a set of $n \geq k$ vector data points into k clusters. In our application the components of the vector include by default the current position and velocity of pedestrians, but may also include acceleration and other metrics over the traces (see Section 4.3). Although the k-means problem is computationally hard, there are good heuristics that make it expedient for our online classification application (e.g., the k-means++ algorithm of the Apache Commons Math library[1]). The algorithm is already well used in the context of motion planning (e.g., [27]).

The algorithm fist 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 a problem-specific notion of distance. Update steps re-calculate the means of the clusters. The 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.

To find an optimal group abstraction we first define a distance function over the data point vector space and specify a maximum distance δ for any two points to be considered close. We do not know in advance the optimal number of clusters (i.e., the optimal value of k), so we explore different values using an efficient strategy. We know from multiple recent studies [6, 20, 7, 9, 1] that 90% of pedestrian groups contain three or fewer people, while there is a computational advantage to simulate the SFM with larger (i.e., fewer) clusters. We therefore set the initial value of k to the number of pedestrians divided by two (implying the most popular average group size, 2, and being close to the maximum possible value of k) and apply the k-means algorithm. If no within-cluster distance is greater than δ, we decrease k and try again. If a within-cluster distance is ever greater than δ we increase k. By repeatedly dividing the plausible range of values we eventually find the smallest value of k that ensures no within-cluster distance is greater than δ.

Although the standard k-means algorithm partitions data points into Voronoi cells, which are disjoint in the vector space of the data, in our application the resulting groups are projected onto Euclidean space and may physically overlap. This arises, for example, when two groups walking in opposite directions pass through each other (see the “passing through” template in Section 4.3). 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. To do this we reduce the repulsive social force between the groups ($\mathbf{f}^{\mathrm{soc}}$ in (2)) and use the physical component ($\mathbf{f}^{\mathrm{ph}}$ 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 (τ in (1)) is also likely to be greater for groups, although some of the latency may be accounted for by the increased mass. Future work will consider the shape and geometry of groups, which could be significantly irregular.

Figure 3 illustrates a scenario of groups diverging and coalescing over time, as interpreted by the group abstraction. The groups labelled 1, 2 and 3 are assumed to comprise a single pedestrian. Group four 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.

Finally, we observe that simulating small, natural groups may be both more efficient and more accurate than simulating individuals.

[1] commons.apache.org/proper/commons-math/

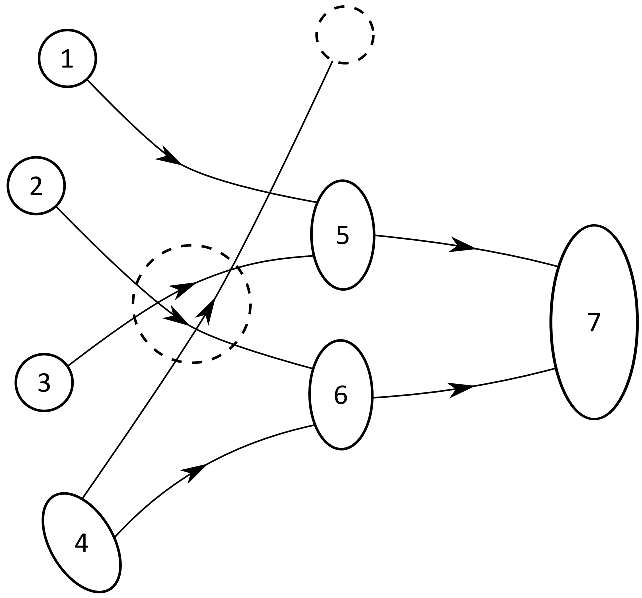

Figure 3: Groups diverging and coalescing.

4.3 Behaviour interpretation

The ACANTO motion planner will make use of statistical model checking (SMC) technology to quantify the probability that hypothesised future trajectories satisfy the requirements of an activity and the users' constraints. The same technology may also be used to calculate behavioural metrics over past traces to classify them. The planner will extend and adapt the PLASMA-lab SMC library[2], which checks properties of traces expressed in bounded linear temporal logic (BLTL). This logic can be used to construct complex behavioural properties that express nested temporal causality. Any language developed to represent or verify activities in Tasks 2.3 and 5.3 is therefore likely to be underpinned by some variant of BLTL.

For reference, the typical abstract syntax of a BLTL property is as follows:

$$\phi = \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \mathrm{F}_{\leq t}\phi \mid \mathrm{G}_{\leq t}\phi \mid \phi\, \mathrm{U}_{\leq t}\phi \mid \mathrm{X}_{\leq t}\phi \mid \alpha \tag{3}$$

$\vee, \wedge$ and $\neg$ are the standard logical connectives *or*, *and* and *not*, while α is an atomic property that is *true* or *false* in a given state. In practice, α 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: $\mathrm{X}_{\leq t}\phi$ asserts that ϕ will be true at time / step t. F is the *finally* or *eventually* operator: $\mathrm{F}_{\leq t}\phi$ asserts that ϕ will be true at some time within t time units. G is the *globally* or *always* operator: $\mathrm{G}_{\leq t}\phi$ asserts that ϕ will be true at all times within t time units. U is the *until* operator: $\psi \mathrm{U}_{\leq t}\phi$ asserts that ϕ will be true within t time units and ψ will be true until it is.

Execution of the SFM requires the consideration of all pair-wise distances between pedestrians. k-means clustering identifies those pedestrians who are physically close and whose behaviour is similar. Treating such clusters as graphs, we may calculate a number of graph-theoretic and other metrics with minimal additional computational cost. For example, notions of group density and cohesion obtained from various centrality and clustering metrics (see, e.g., [21, Chap. 7]) are useful to predict whether groups will pass by or pass *through* each other and to identify key (literally pivotal) pedestrians.

[2]project.inria.fr/plasma-lab

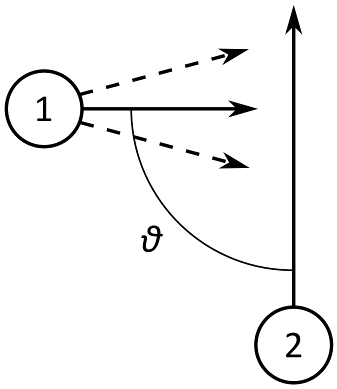

Figure 4: “Give way” behavioural template.

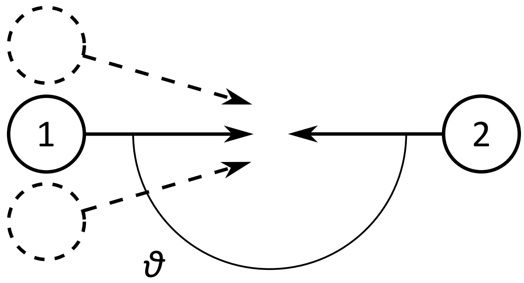

Figure 5: “Head on” behavioural template.

Detecting behavioural templates

We now describe the simple momentum based identification of some behavioural templates that are especially relevant for therapeutic group activities. The templates we consider are intuitively described as “giving way”, “head on” and “passing through”. Giving way occurs when one group slows to allow another to pass. Head on occurs when two groups deviate to avoid a head on collision. Passing through is when the members of colliding groups choose to pass between the members of the other group. To detect these templates we make a *linear* projection of group motion to identify potential collisions. We do not use the SFM in this instance precisely because it will tend to avoid the collisions. We thus anticipate the collisions that the pedestrians themselves will be anticipating.

Figure 4 illustrates the scenario where group 1 gives way to group 2. This template requires the collision angle ϑ to be $90^\circ \pm \omega$ ($\omega = 15^\circ$ is depicted by dashed lines in the figure). Beyond ω we assume that either another template will apply or the standard smooth dynamics of the SFM will take over. In terms of momentum, we require that group 2 is bigger and/or moving faster than group 1. As a future refinement, the momentum requirements may be superseded if it is possible to infer that group 2 has social precedence over group 1, e.g., if group 2 are elderly and group 1 are not.

Figure 5 specifically illustrates the “head on” template, but is also illustrative of the pre-conditions for “passing through”. This template requires the collision angle ϑ to be $180^\circ \pm \omega$ ($\omega = 15^\circ$ is depicted in the figure). The precise nature of the template depends on the relative masses and densities of the the two groups. If the two groups are relatively dense we expect to see mutual deviation in proportion to mass (“head on”). If either or both of the groups are sparse, the groups may pass through each other (“passing through”).

Finally in this section we note that queueing behaviour is a complex sequence of stops and starts that is not well modelled by smooth dynamics. Although people queueing in the environment may affect the dynamics of those moving more actively, queueing is less relevant to group motion and we therefore leave its implementation (along with other individual-based templates) to future work.

5 Prospects

We have designed a novel hierarchical approach to solve a complex dynamic motion planning problem and have made a prototype implementation in Java, using the Apache Commons Math library[1] to provide standard k-means algorithms and ordinary differential equation solvers (for the SFM). We choose Java for its convenience and for its compatibility with other prototyping environments, such as Matlab. Java is also likely to be the programming language for the final implementation, using the Android operating system. Note that we have already demonstrated that Java is adequately fast for this type of application [4], while the computational

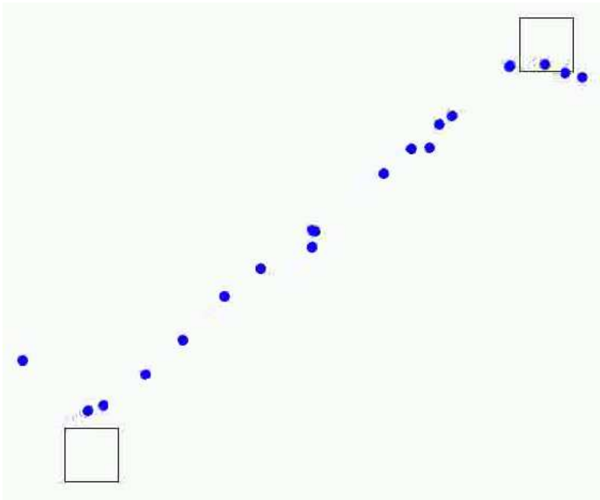

Figure 6: Independent pedestrians.

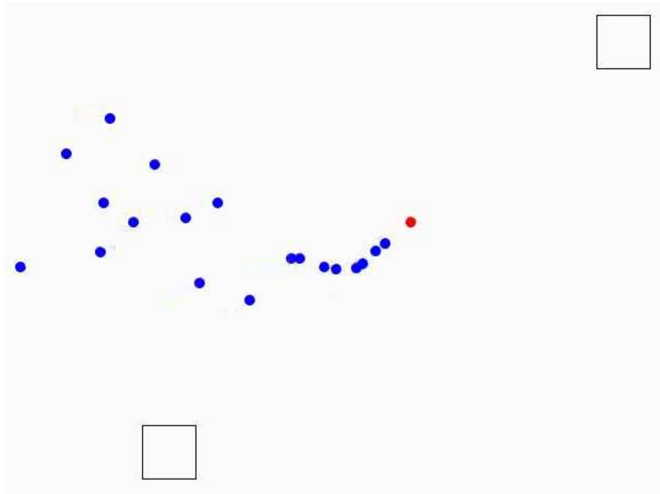

Figure 7: "Follow the leader."

complexity we face in ACANTO cannot simply be addressed by the small constant factor gains of using a lower level language.

The key benefits of our framework are its efficiency and its ability to take advantage of available data and computing power. We have tested our ideas on a variety of simulated scenarios, but it remains for us to apply the system to the complexity and unpredictability of real pedestrians. Since such tests cannot hope to be exhaustive, an intermediate goal is to test our framework on highly complex simulated scenarios that include the noise and uncertainty that we anticipate in reality.

Although the current deliverable is principally concerned with *detecting* groups, we have also used our implementation to investigate the parameters necessary to *create* groups, as will be required for activities. In particular, we have considered the problem of maintaining group cohesion when members of the group have different abilities, i.e., different preferred speeds of moving. A priori it seems plausible to use the SFM's attractive forces ($\mathbf{f}^{\mathrm{att}}$ in (1)) to keep the group together. In practice, however, these forces merely model the *desire* to stay together.

Figures 6 and 7 show frames from simulations that illustrate two of many considered problematic scenarios. In both figures the pedestrians (dots) are attempting to visit the two locations denoted by squares and have a range of desired speeds with a maximum to minimum ratio of 2. Figure 6 illustrates the case where the pedestrians act independently and visit the locations as fast as they can. The result is that there is no group cohesion. Figure 7 illustrates the case where the pedestrians try to keep up with the fastest member of the group (indicated in red). The figure shows how the slow pedestrians must skip the first location to stay with the group.

Our work on maintaining group cohesion continues, however we anticipate that cohesion will not be achieved by merely adjusting the parameters of the SFM (although this may form part of the solution). Since we cannot reasonably speed up slow pedestrians, we believe the best approach will be to enforce pauses that allow slow pedestrians to catch up, while avoiding the potential frustration to faster pedestrians who might otherwise be forced to slow down.

Finally, we briefly itemise below some other future directions of our ongoing research.

1. Exploitation of the shape, geometry and structure of groups to better predict, guide and plan group motion.
2. Integration of social factors into template detection (e.g., social precedence of giving way).
3. Implementation of low (i.e., individual) level template detection (e.g., queueing).

4. Incorporation of advanced features of human motion models identified by D2.1.1.
5. Integration of local networking to identify other users and members of an activity group.

Bibliography

[1] Stefania Bandini, Andrea Gorrini, and Giuseppe Vizzari. Towards an integrated approach to crowd analysis and crowd synthesis: A case study and first results. *Pattern Recognition Letters*, 44:16–29, 2014.

[2] Favyen Bastani, Yan Huang, Xing Xie, and Jason W. Powell. A greener transportation mode: Flexible routes discovery from gps trajectory data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 405–408, New York, NY, USA, 2011. ACM.

[3] Chih-Wei Chen, Rodrigo Cilla Ugarte, Chen Wu, and Hamid Aghajan. Discovering social interactions in real work environments. In *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pages 933–938. IEEE, 2011.

[4] A. Colombo, D. Fontanelli, D. Gandhi, A. De Angeli, L. Palopoli, S. Sedwards, and A. Legay. Behavioural templates improve robot motion planning with social force model in human environments. In *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–6, Sept 2013.

[5] 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 *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3602–3608, Dec 2013.

[6] Marco Costa. Interpersonal distances in group walking. *Journal of Nonverbal Behavior*, 34(1):15–26, 2010.

[7] Mizar Luca Federici, Andrea Gorrini, Lorenza Manenti, and Giuseppe Vizzari. Data collection for modeling and simulation: case study at the university of Milan-Bicocca. In *Cellular Automata*, pages 699–708. Springer, 2012.

[8] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *Robotics, IEEE Transactions on*, 21(6):1077–1091, 2005.

[9] Natalie Fridman, Gal A Kaminka, and Avishay Zilka. The impact of culture on crowd dynamics: an empirical approach. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 143–150. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[10] Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 63–72, New York, NY, USA, 1999. ACM.

[11] Maxime Gariel, Ashok N Srivastava, and Eric Feron. Trajectory clustering and an application to airspace monitoring. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1511–1524, 2011.

[12] Dirk Helbing, Illés Farkas, Péter Molnár, and Tamás Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. In Michael Schreckenberg and Som Deo Sharma, editors, *Pedestrian and Evacuation Dynamics*. Springer, 2002.

[13] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.

[14] Dirk Helbing, Illés J. Farkas, and Tamás. Vicsek. Freezing by heating in a driven mesoscopic system. *Phys. Rev. Lett.*, 84:1240–1243, 2000.

[15] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.

[16] John James. The distribution of free-forming small group size. *American Sociological Review*, 1953.

[17] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 593–604, New York, NY, USA, 2007. ACM.

[18] James 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.

[19] Brendan Morris and Mohan Trivedi. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[20] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4):e10047, April 2010.

[21] Mark Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[22] Claudio Piciarelli, Gian Luca Foresti, and Lauro Snidaro. Trajectory clustering and its applications for video surveillance. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 40–45. IEEE, 2005.

[23] S. Raupp Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *Visualization and Computer Graphics, IEEE Transactions on*, 7(2):152–164, April 2001.

[24] Francisco Arturo Rojas, Jin Hyoung Park, and Hyun Seung Yang. Group agent-based steering for the realistic corner turning and group movement of pedestrians in a crowd simulation. *Proceedings of Computer Animation and Social Agents (CASA 2013)*, 2013.

[25] Francisco Arturo Rojas and Hyun Seung Yang. Immersive human-in-the-loop hmd evaluation of dynamic group behavior in a pedestrian crowd simulation that uses group agent-based steering. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 31–40. ACM, 2013.

[26] Francisco Arturo Rojas and Hyun Seung Yang. Minimizing collision among social groups in wide-open spaces. In *2014 International Conference on Cyberworlds (CW)*, pages 77–84. IEEE, 2014.

[27] C. Sung, D. Feldman, and D. Rus. Trajectory clustering for motion prediction. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1547–1552, Oct 2012.