Gaussian Mixture Based Motion Prediction for Cluster Groups of Mobile Agents ⋆

Anegi James * **Efstathios Bakolas** *

* *University of Texas at Austin, Austin, TX 78705 USA (e-mail: anegij@utexas.edu, bakolas@austin.utexas.edu).*

Abstract: This paper presents a Gaussian mixture model based method for estimation of multiple mobile agent trajectories over time following a social force motion model. From an initial observation, groups of agents are formed using a bottom-up agglomerate clustering approach. Each cluster constitutes a component of the Gaussian mixture model, with mean position of the cluster members. An Adaptive Gaussian Sum filter estimates the future trajectories of the agent clusters using the Gaussian mixture model input. The proposed algorithm is capable of reducing the computation time compared to the standard Adaptive Gaussian Sum filter and provides solutions with lower variance due to utilization of agent groups. The efficacy of the proposed method is demonstrated via extensive numerical simulations.

Keywords: Adaptive Gaussian Sum Filter, Gaussian Mixture model, Motion prediction, Clustering

1. INTRODUCTION

Motion prediction is an important component to enable safe motion planning for autonomous mobile agents. An agent (by agents we refer to robots, autonomous or manned vehicles, humans etc.) in a field populated by dynamic and static obstacles needs dependable information about the future planned motion of the dynamic obstacles. An accurate prediction assists the agent to identify more efficient paths in its vicinity with a greater likelihood of avoiding collisions and avoid computing infeasible paths leading to freezing Trautman and Krause (2010). Collision avoidance requires some understanding of an agent's future position with a probabilistic variance to account for the uncertainty in the estimate. Unexpected and sudden changes in the paths of agents and incomplete information about the mobile agent may result in inconsistencies between estimate and reality. In many cases the intent of the agent may be unknown and is also likely to vary between different agents. The main aim of this work is identifying the future motion paths for mobile agents in the field of view of an autonomous system with minimal information about the agents apart from the current state/short previous time history of the state. An Adaptive Gaussian Sum filter based approach is used to predict the future states by identifying cluster trends present in the observed states.

Literature Review: Several methods for motion prediction have been developed for several applications in robotics, target tracking, aerospace etc. In Brown and Rogers (2017) and Schulz et al. (2018), a Monte Carlo based approach is used to quantify the predicted path uncertainty. Hu et al. (2007) and Zheng et al. (2021) use density-based estimation methods to enable detection of future paths for large-scale multi-agent systems. In Liu et al. (2017), reachability sets of all possible states are identified for a kinematic model representing pedestrian agents. In Fridovich-Keil et al. (2020) a confidence aware human prediction method, where the human motion is assumed to try to maximize a reward function, is proposed and the reliability of the result depends on the confidence of the prediction. In Patterson et al. (2019) a probabilistic representation of the agent's future position is found by identifying polynomials to represent the uncertainty region of predicted paths using Gaussian process regression. Aoude et al. (2013) also uses a Gaussian process framework for prediction of dynamic obstacle paths. Several authors Yokoyama (2018), Park et al. (2016), Park et al. (2019), Schulz et al. (2018) and Škovierová et al. (2018) identify the intents of the agents for different contexts and refine the predicted motion problem to account for these intents.

Gaussian processes are widely used for predicting the path of mobile agents Patterson et al. (2019). However, a single Gaussian process can only encode information about a single agent while a Gaussian mixture model can encode information about multiple agents. In Terejanu et al. (2011), an Adaptive Gaussian Sum filter has been developed which not only uses new information to update the Gaussian mixture but also updates the weights in time. Data clustering has been extensively studied and various methods exist to accomplish this task including partitioning, hierarchical, density-based (Ester et al. (1996)), grid based, model based, pattern based etc approaches, trajectory clustering (Bian et al. (2018), Chen et al. (2011), Atev et al. (2010)). However, most methods require an apriori fixed number of clusters throughout the prediction horizon, which is undesirable for the highly variant scenarios for agent motion prediction. Bera et al. (2016) uses an agglomerative hierarchical clustering approach without initial cluster number information. Li et al. (2004), Jensen et al. (2007) and Jeung et al. (2010) implement a method for clustering moving objects based on a short history of the agents motion allowing dynamic clusters as well as groups that are likely to move together to be formed.

Main Contributions: In this work, we consider the motion prediction problem for large groups of agents taking into account their likeliness to move together in smaller groups. We assume access to an initial observation of each agent's position and velocity and observations of their state at specified intervals. The closeness between agents is exploited to cluster them into groups that are assumed to

⋆ This research has been supported in part by NSF award ECCS-1924790.

move closely together for at least a portion of the simulation time. The motion of the agent clusters is predicted using an Adaptive Gaussian Sum filter which propagates the states to a future time incorporating clusters of agents as components of the Gaussian mixture model. The variance of the estimate is also predicted by the filter giving a probabilistic prediction for the position occupied by the agents. The main aim of this work is to provide a new algorithmic framework for motion prediction of dynamic agents using a Gaussian sum filter setup while reducing computation time and (large) uncertainties by making use of the relation between paths of agents. *Outline:* In Section II we introduce and set up the background information required for the motion prediction problem; the Adaptive Gaussian Sum filter and the hierarchical agglomerative clustering technique for identifying agent groups. Section III outlines the problem definition and proposed algorithm based on elements from Section II. Section IV provides simulation results using this proposed algorithm with relevant examples and scenarios. Section V concludes the paper with final remarks and future steps.

2. PRELIMINARIES

This section introduces all the background relevant to the proposed solution.

2.1 Adaptive Gaussian Sum Filter

Consider a group of N agents moving in a known domain. The state of agent i in the group is given by $x^i = [p_x^i, v_x^i, p_y^i, v_y^i]^{\mathrm{T}}$ with $[p_x^i, p_y^i]^{\mathrm{T}}$ representing its position and $[v_x^i, v_y^i]^{\mathrm{T}}$ its velocity in the 2-dimensional space. The state x^i for $i = 1 : N$ is a Gaussian random vector with mean $\mu^i \in \mathbb{R}^4$, covariance matrix $P^i = {P^i}^{\mathrm{T}} > 0$ and corresponding PDF, $\rho_{\mathcal{N}}(x^i; \mu^i, P^i) = (1/\sqrt{(2\pi)^4|P^i|})\mathrm{e}^{-\frac{1}{2}(x^i-\mu^i)^T {P^i}^{-1}(x^i-\mu^i)}$. Given an initial configuration $x_0 = [{(x^1)}^{\mathrm{T}} \dots {(x^N)}^{\mathrm{T}}]^{\mathrm{T}}$ for a group of N agents, the Gaussian mixture model for the group is initialized with N components; one for each agent. Each component of the Gaussian mixture follows a Gaussian probability distribution. The probability density of the mixture model associated with the state vector $x = [{(x^1)}^{\mathrm{T}} \dots {(x^N)}^{\mathrm{T}}]^{\mathrm{T}}$ at time t is represented by $p(x(t)) = \sum_{i=1}^N w_t^i \rho_{\mathcal{N}}(x(t); \mu_t^i, P_t^i)$ where μ_t^i and P_t^i are the state mean and state covariance of agent i at time t. w_t^i is the weight of each component of the Gaussian mixture at time t, initially set to a common value for all $i = 1 : N$. The weights are selected such that $w_0^i = 1/N$ for $t = 0$, $w^i \geq 0$ and $\sum_{i=1}^N w^i = 1$ for $i = 1 : N$ and $\forall t \geq 0$. The resulting mixture is propagated using the Adaptive Gaussian Sum Filter (AGSF) described next. Assume each of the N agents follow the discrete-time dynamical model governed by the following stochastic differential equation and measurement model:

$$x_{k+1}^i = f(t_k, x_k^i) + \eta_k^i \qquad z_k^i = h(t_k, x_k^i) + v_k^i \qquad (1)$$

where $x_k^i \in \mathbb{R}^n$ represents the state vector at time t_k and $\eta_k^i \in \mathbb{R}^n$ represents the zero mean white Gaussian noise with covariance function $Q_k \geq 0 \in \mathbb{R}^{n\times n}$. The measurements z_k^i assume to follow (1) with the non-linear function h representing the measurement model and zero mean white Gaussian noise $v_k^i \in \mathbb{R}^n$ with covariance $R_k \geq 0 \in \mathbb{R}^{n\times n}$. The measurement noise v_k is assumed to be uncorrelated and η_k^i and v_k^i are independent of each other, that is, $\mathbb{E}[\eta_k^i (v_k^i)^{\mathrm{T}}] = 0$. Let $Z_k = [(z_k^1)^{\mathrm{T}} \dots (z_k^N)^{\mathrm{T}}]^{\mathrm{T}}$ for $k = 1 : T_o$ be the vector of measurements of the states of the N agents for some observation time T_o. The conditional PDF for the Gaussian mixture given state measurements Z_k is given by a weighted finite sum of the individual Gaussian components:

$$p(t_{k+1}, x_{k+1}|Z_k) = \sum_{i=1}^N w_{k+1}^i \rho_{\mathcal{N}}(x_{k+1}; \mu_{k+1|k}^i, P_{k+1|k}^i)$$

where $w_{k+1|k}^i, \mu_{k+1|k}^i$ and $P_{k+1|k}^i$ represent the weight, mean and covariance of the i^{th} Gaussian component. The weights are constrained to positivity, $w_k^i > 0$ and unity, $\sum_i^N w_k^i = 1$. The filter estimates the mean and covariance for each component according to the conventional Extended Kalman Filter time update equations in (2):

$$\mu_{k+1|k}^i = f(k, \mu_{k|k}^i), \qquad P_{k+1|k}^i = A_k^i P_{k|k}^i {A_k^i}^{\mathrm{T}} + Q_k, \quad (2)$$

where $A_k^i = \frac{\partial f(k, x_k^i)}{\partial x_k}\Big|_{\mu_{k|k}^i}$. The measurement update follows (3) using Bayes' rule:

$$\begin{aligned} \mu_{k+1|k+1}^i &= \mu_{k+1|k}^i + K_k^i(z_k^i - h(t_k, \mu_{k+1|k}^i)) \\ K_k^i &= P_{k+1|k}^i (H_k^i)^{\mathrm{T}} (H_k^i P_{k+1|k}^i (H_k^i)^{\mathrm{T}} + R_k)^{-1} \\ P_{k+1|k+1}^i &= (I - K_k^i H_k^i) P_{k+1|k}^i \\ H_k^i &= h(t_k, x_k^i) \\ w_{k+1|k+1}^i &= \frac{w_{k+1|k}^i \beta_k^i}{\sum_{i=1}^N w_{k+1|k}^i \beta_k^i} \\ \beta_k^i &= \mathcal{N}(z_k^i - h(t_k, \mu_{k+1|k}^i), H_k^i P_{k+1|k}^i (H_k^i)^{\mathrm{T}} + R_k) \end{aligned} \qquad (3)$$

The evolution of the true PDF of the agent states (x_k) obeys the Chapman-Kolmogorov partial differential equation. The time update for the component weights minimizes the square difference between the true PDF and the Gaussian mixture approximation:

$$\begin{aligned} \min_{w_{k+1}} J &= \frac{1}{2} w_{k+1}^{\mathrm{T}} M w_{k+1} - w_{k+1}^{\mathrm{T}} N w_k \\ \text{s.t} \quad & 1^{\mathrm{T}} w_{k+1} = 1, w_{k+1} \geq 0 \\ M_{ij} &= \mathcal{N}(\mu_{k+1}^i - \mu_{k+1}^j, P_{k+1}^i + P_{k+1}^j) \\ N_{ij} &= \int \mathcal{N}(f(t_k, x_k); \mu_{k+1}^i, P_{k+1}^i + Q_k) \\ & \quad \times \mathcal{N}(x_k; \mu_k^j, P_k^j) dx_k \end{aligned} \qquad (4)$$

The goal of the optimization problem given in (4) is to find the optimal weights of the GM components to account for changes in the distribution over time. The resulting PDF for each agent is calculated using (1) from the results of (2)-(4) (also refer to Terejanu et al. (2011)).

2.2 Clustering

The AGSF, introduced in Section 2.1, permits the use of a Gaussian mixture to propagate the state position and velocity for groups of agents that are correlated. This section details the clustering process used to identify the number and composition of the components of the Gaussian mixture. An agglomerate clustering algorithm is applied to determine agent groups based on various crowd features such as closeness in distance and velocity. To this aim, consider two agents i and j with states x^i and x^j in close proximity, with similar features including speed and orientation. The position, velocity and direction of motion of an arbitrary agent i is represented by p^i, v^i and ϕ^i

respectively. Two agents are said to be related if all of the following conditions hold at arbitrary time t:

C1: Distance between agents is small $\|p_t^i - p_t^j\|_2 < \tau_s$
C2: Difference in speed is small $\|v_t^i - v_t^j\|_2 < \tau_v$
C3: Agents have similar orientation $|\phi_t^i - \phi_t^j| < \tau_a$

where $\tau_s > 0$, $\tau_a > 0$ and $\tau_v > 0$ represent thresholds for relative closeness in distance, angular orientation and velocity of the two agents. The correlation value, $\rho_{i,j} = (1/T_0)\sum_t^{T_0} \delta_t(i,j)$, for a pair of agents determines whether two agents are assumed to move together as a cluster if they are related for a majority of the observation time.

$$\begin{aligned} &\delta_t(i,j) = 1 \quad \text{if agents are related at time } t \\ &\delta_t(i,j) = 0 \quad \text{when conditions not satisfied} \end{aligned} \tag{5}$$

Eq.(5) sets the correlation value $\rho_{i,j}$ for the pair of agents if conditions **(C1-C3)** are satisfied for the initial observed state time series data $\forall t \in [0, T_0]$. The correlation value is identified for every pair among the N agents and initial clusters are determined by grouping strongly correlated agents. This provides clusters of pairs of agents (x^i, x^j) and individual agents $\{x^i\}$ forming their own cluster.

For further clustering after identifying correlated pairs, a bottom-up hierarchical approach is used to group the agent clusters. Each cluster is progressively merged by analysing their geometric proximity and velocity orientation. The closeness between individual agents is measured in terms of the Euclidean distance while the Hausdorff distance metric measures the relative closeness of clusters. The Hausdorff distance for two non-empty subsets of a metric space is given by

$$d_H(X,Y) = \max\{\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(X,y)\},$$

where $d(x,Y) = \inf_{x\in X} d(x,y)$.

Each agent in the cluster is assumed to comprise a vertice of a connectivity graph such that the number of agents in the cluster N_c is equivalent to the number of vertices. If two vertices (agents i and j with vertices v_i and v_j of the graph $\mathcal{G}$) are closely correlated, they have an edge between them. A cluster with N_c agents has e_c edges since the vertices in a cluster are correlated. If two clusters have a greater than threshold number of edges, then they are said to be *correlated* and merge to form one larger cluster. The graph density is measured based on the intra-cluster proximity. For a cluster with $k-1$ vertices, the number of edges that should be prevalent when adding an extra member to the cluster is denoted by $\hat{e}_k$ and defined as:

$$\hat{e}_k = \begin{cases} k^2/2 & \text{if } k \text{ is even} \\ (1/2)(k-1)\big(1+(1/2)(k-1)\big) & \text{if } k \text{ is odd} \end{cases} \tag{6}$$

If the inequality

$$e_{p+q} \geq \hat{e}_{p+q} + (e_p - \hat{e}_p + e_q - \hat{e}_q)$$

is satisfied, then cluster p and q are correlated and should be merged into one cluster with $n_p + n_q$ agents. Figure 1 illustrates an example of the clustering process for two agent cluster groups with more than a single agent in each cluster. Cluster p has 3 vertices with 2 edges between them and cluster q has 2 vertices with 1 edge between them. There is a potential edge (orange) between vertex k of cluster p and vertices j and l of cluster q. The clusters are merged if the number of edges present in the merged cluster is above the threshold number of edges required.

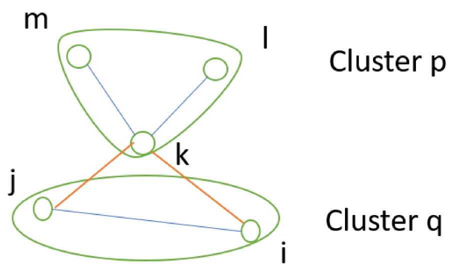


Fig. 1. Illustration of clustering process for two agent clusters with $n > 1$ number of agents in each cluster. The blue lines represent the current edges between vertices in the respective cluster whereas the orange lines indicate the potential edges between elements of clusters p and q.

3. PROBLEM FORMULATION

In this section, we formalize the dynamics of the agents and clusters and the method applied for state estimation.

3.1 Agent Dynamics

Let $x^i = [p_x^i\ v_x^i\ p_y^i\ v_y^i]^{\mathrm{T}}$ represent the state of an arbitrary agent i in a 2-dimensional frame of reference. The problem is formulated with a double integrator model representing the motion model for each agent i given by $\ddot{p}_x(t) = u_x(t)$ and $\ddot{p}_y = u_y(t)$. The input $u_x(t)$ and $u_y(t)$ of the double integrator model is determined by a variation of the social force model provided in Section 3.3. The acceleration of the agent is given by $\ddot{p}_x = F_x$ and $\ddot{p}_y = F_y$ for both x and y directions which also defines the input $u(t)$ to the system. This social force models the interaction between agents and their environment ensuring that the agents avoid collisions in the predicted path.

3.2 Cluster Dynamics

Let C represent a cluster of agents which may be comprised of multiple agents moving as a group or an individual agent constituting a cluster group. The state vector of the cluster $c = [p_x^c\ v_x^c\ p_y^c\ v_y^c]^{\mathrm{T}}$ provides similar position and velocity information as with the individual agent case but for the cluster as a whole. For a given cluster comprised of n number of agents, the cluster state c is given by the mean of the states of each individual agent in the cluster, $c = \frac{1}{n}\sum_{i=1}^{n} x^i$. The covariance for the cluster is initially set to diagonal matrix, $\Sigma_c = diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$, where $\sigma_i > 0$, for $i = 1:4$, are the variances of each state component and is updated over time using the covariance update equation described in (3). The dynamic model used for clusters is also given by the agent dynamic model where the acceleration is modelled using a social force model to account for the interactions between agents and clusters, which is described in detail next.

3.3 Social Force Model

The dynamics of the motion model for the moving agent uses the social force model to identify the input acceleration to the system which affects the prediction of how the agent moves. The social force model applied is a variation of that proposed by Helbing and Molnár (1995), where the force acting on an agent depends on three main effects: the interaction/repulsion between agents, the desire of the agent and the interaction with static obstacles in the agent's path.

The formulation for the social force model is given below for the force acting on an agent A with state x_A interacting with another agent B with state x_B and a stationary obstacle O which is located at (x_O, y_O). For a given agent A, provided information about its intent location I and desired speed v_A^0 with which it attempts to reach its goal, we define the destination force F_A^{des} as follows:

$$e_A(t) := \frac{r_A - r_A(t)}{\|r_A^k - r_A(t)\|} \quad F^{des} = \frac{1}{\tau}(v_A^0 e_A - v_A), \qquad (7)$$

where $r_A(t)$ denotes the actual position of A, r_A^k is the desired location, e_A is the desired direction to be taken towards the goal and v_A is the actual velocity of A at the current time step. The interaction of agent A with another agent B is given by the interaction force, F^{int}, between them which is a repulsive force ensuring they do not collide during motion.

$$\begin{aligned} F^{int} = A_{int} \exp \frac{d_{AB}}{B_{int}} + k_1 \max(d_{AB}, 0) n_{AB} \\ + k_2 \max(d_{AB}, 0) \Delta v_{AB} t_{AB} \end{aligned} \qquad (8)$$

where A_{int} and B_{int} are constant parameters of the interaction force, d_{AB} is the effective distance between A and B, n_{AB} is the unit normal in the direction from A to B, t_{AB} is the unit perpendicular to n_{AB} and Δv_{AB} is the difference in velocity between A and B. The repulsion force between agent A and a static obstacle in its path O, is modeled by the obstacle force, F^{obs}, which is defined as:

$$\begin{aligned} F^{obs} = A_{obs} \exp \frac{R_A - d_{AO}}{B_{obs}} + k_1 \max(R_A - d_{AO}, 0) n_{AO} \\ + k_2 \max(R_A - d_{AO}, 0) v_A t_{AO} \end{aligned} \qquad (9)$$

where A_{obs} and B_{obs} are constant parameters of the interaction force, R_A is the radius of the agent, d_{AO} is the distance between the agent and the obstacle, n_{AO} is the unit normal in the direction from A to O, t_{AO} is the unit perpendicular to n_{AO} and v_A is the velocity of agent A. The net force on an agent A as a result of every other agent and obstacle in the field of view is given by the sum of forces, $F_A = \sum_{A \neq j, j=1}^{N} F_A^j = F_A^{des} + F_A^{int} + F_A^{obs}$.

3.4 Problem Definition

Let a group of N agents in motion in an $M \times L$ domain with states denoted by x_i for $i = 1 : N$ be given and let the field of view be a bird's eye view of the domain. Each agent follows the dynamic model described in Section 3.1. Time series data of state observations for all N agents is available for time $t = t_0$. The PDF of the clustered agent distribution is initially given by $p(t_0, x(t_0)) = \sum_{i=1}^{N_c} w_{t_0}^i \rho_{\mathcal{N}}(x(t_0); \mu_{t_0}^i, P_{t_0}^i)$. The objective of this work is to predict the density evolution of the agent clusters. The initial state observations providing position and velocity data for the agents are used to group agents into appropriate clusters that capture agent motion patterns. Further the motion of the agent clusters are predicted using the AGSF which propagates the state estimate of the agent clusters to future time steps. The main aim is to predict the future PDF of the clusters from the initial time to the final simulation time t_f.

Measurements are taken at regular intervals to reduce uncertainty. The filter provides estimates of the cluster mean state and its covariance. The covariance indicates the potential area occupied by the cluster due to uncertainty in the estimate attributed to various factors such as measurement and modeling errors. The posterior PDF distribution of agent states is a Gaussian mixture representing the agent cluster position and velocity estimates at time t_f using initial observed data which is given by $p(t_f, x(t_f)) = \sum_{i=1}^{N_c} w_{t_f}^i \rho_{\mathcal{N}}(x(t_f); \mu_{t_f}^i, P_{t_f}^i)$. We identify the PDF depicting the position of the agent in the domain for the entire time span of the simulation ($t \in [t_0, t_f]$). This provides a state estimate of every agent cluster at each time across the span and an overall density estimate of the occupied area in the domain.

4. PROPOSED SOLUTION APPROACH

Consider a group of N agents, each characterised by an initial observed state x_0^i for $i = 1 : N$. The initial observation provides information about the position as well as the velocity of each agent. This information is used by the clustering algorithm to initiate clusters based on position, orientation and speed which provides N_c number of clusters where $N_c \leq N$. An agent may be placed into a cluster comprising of only itself (in which case the cluster reduces to a singleton) or may be grouped with other agents that are estimated to move together in the future.

For clusters comprising of a group of agents, the group dynamics follow the cluster dynamics given in Section 3.2. The mean position and velocity of the group define the overall position and speed attributed to the cluster. It is also assumed that once the clusters are defined, new clusters will not be formed. This initial distribution of agents across clusters provides C_i clusters where $i = 1 : N_c$. Propagation of mean and covariance of the clusters is performed using the Adaptive Gaussian Sum Filter detailed in Section 2.1. The AGSF is initialized with a Gaussian mixture with N_c components, one for each cluster. The filter propagates the states to a future time using the motion model described in Section 3. The periodic measurement update of the AGSF allows the predicted states to be more accurate, have a lower variance, and correct for errors that may occur due to lack of information about agent intent. The measurement update is implemented when the variance of the system grows above a threshold, $\tau_{cov} \geq 3\sigma_x^{1/2}$, thereby managing the variance of the prediction from growing out of bounds. Additionally, the regular measurement update allows any differences in the cluster composition to be identified and corrected if two agents that were initially moving together separate their paths at a future time.

Algorithm 1 outlines the basic steps in the trajectory estimation process for the group of N moving agents. In step 1, the initial state of the system is set to the observed state at time t_0. In steps 2-5, the agents are clustered (Section 2.2) and the cluster parameters are identified for the Gaussian mixture model. Steps 6-10 utilize the AGSF from Section 2.1 to find the future estimate of the mean $(x_{T_f}^1 \ldots x_{T_f}^N)$ and covariance for each cluster of agents. The social force model described in Section 3.3 is used to approximate the agents' motion while capturing the interaction between them and their environment (obstacles). The AGSF further calculates the estimate for the future position over the interested time span which is refined using frequent state observations.

5. SIMULATION RESULTS

This section presents the results of using the AGSF with the clustering algorithm and a benchmark without clustering. A maximum grid size of $20 \times 20 [m^2]$ is assumed for the

Algorithm 1 Agent motion prediction algorithm

Require: $N \geq 0$, $x_0^1 \dots x_0^N$, T_f

```
x^i ← x^i_0
Cluster agents into groups C_j
Calculate cluster mean c_j
Find Gaussian mixture model
w_i ← 1/N_c
for t = 1 : T_f do
    Run Adaptive Gaussian Sum Filter
    Update weights
end for
x^1_{T_f} ... x^N_{T_f} ← N(μ, Σ)
```

field of view and movement of the agents. The simulations have a run time of 10 seconds with an increment of 0.1[s]. Each run considers the motion of 20 agents with double integrator dynamics, where the acceleration is determined by the social force model described in Section 3.3 to simulate the interaction between agents and static obstacles.

While the composition of clusters may change for the duration of the simulation, the number of clusters remain a constant from when they are first defined. The observation update for the filter is performed when the covariance of the prediction increases above a threshold, where $\tau_{cov} = 5$.

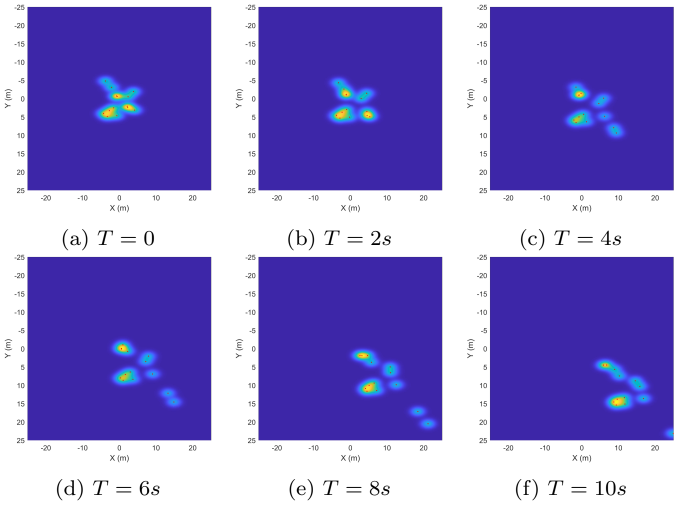
5.1 Comparison of Effect of Clustering

As a benchmark, the predicted motion of the unclustered crowd data is found using the AGSF in Figure 3. The observed data prior to clustering is the initial condition for the simulation. In this case, each individual pedestrian represents a component of the Gaussian mixture distributed with equal weight. The simulation is completed for a 10s time span with 0.1s increments.

The evolution of the PDF, representing the position and variance of the cluster states, is illustrated in Figure 2. Agent clusters represent both clusters composed of individual agents and those with multiple agents comprising one cluster. Each cluster forms a component of the Gaussian mixture and the propagated states represent the mean and covariance of the cluster state as a whole. The red points in Figures 2 and 3 represent the mean position of each cluster while the peaks represent the mean of the associated PDF for each Gaussian component. Both clustered and un-clustered distributions are similar initially but the singleton clusters separate and spread out to a greater degree over time. Table 5.1 indicates the qualitative differences between both cases. It is observed that the total area occupied by the state position estimate and the associated variance is larger for the unclustered case. As a result the available clearance for maneuvering around these agents is tighter. The prediction with clusters retains this grouping information allowing the estimated state positions to be more accurate over time. Figure 4 depicts the comparison between the trajectories generated by both cases. The left plot depicts the clustered case and has fewer agents as well as less variance in the state position estimate. The right plot represents the single agent clusters case and we notice greater variance in the estimate.

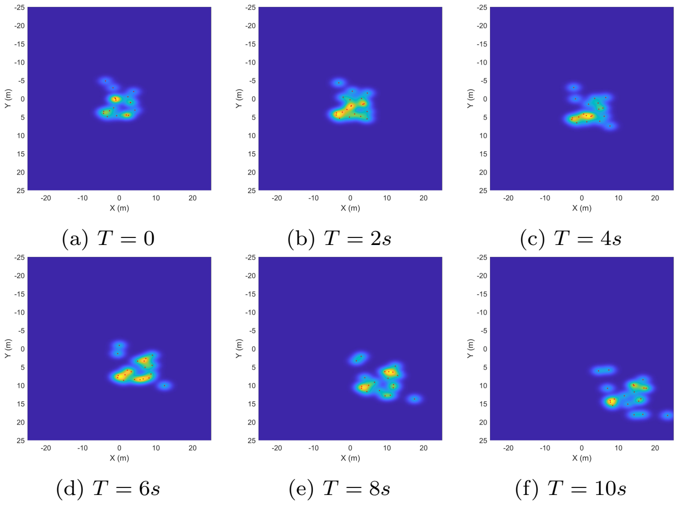
Table 5.1: Clustered vs unclustered results

Type	Simulation Time [s]	Occupied Area [%]
Unclustered	121.43	0.6495
Clustered	93.4	0.4547



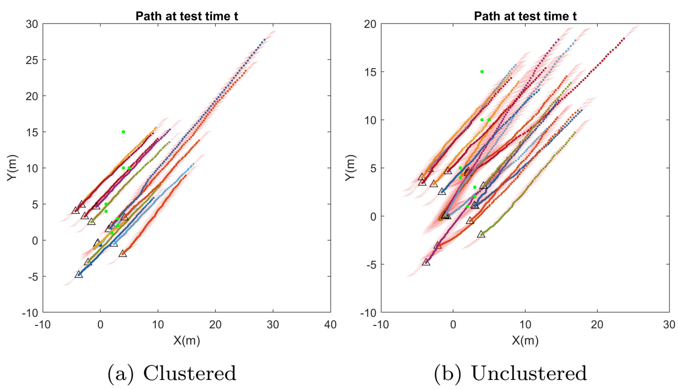
(a) $T = 0$ (b) $T = 2s$ (c) $T = 4s$ (d) $T = 6s$ (e) $T = 8s$ (f) $T = 10s$

Fig. 2. PDF of estimates of future motion (position and variance) of moving agents with clustered centers (red) representing the centers of each group of agents.



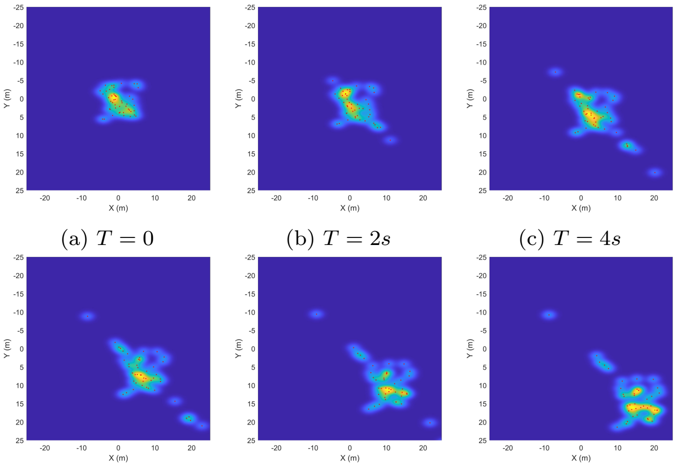
(a) $T = 0$ (b) $T = 2s$ (c) $T = 4s$ (d) $T = 6s$ (e) $T = 8s$ (f) $T = 10s$

Fig. 3. PDF of estimates of future motion (position and variance) of moving agents with centers (red) representing the centers of individual agents.



(a) Clustered (b) Unclustered

Fig. 4. Comparison of predicted path for clustered (left) vs individual (right) agents with uncertainty region (light red) around the predicted path. The static obstacle locations are depicted with green circles and the initial position for each cluster with a triangle.



(a) $T = 0$ (b) $T = 2s$ (c) $T = 4s$

(d) $T = 6s$ (e) $T = 8s$ (f) $T = 10s$

Fig. 5. PDF of estimates of future motion (position and variance) of a large group of moving agents with centers (red) representing the centers of agent clusters.

5.2 Multi-Agent System Scenario

We have also tested our algorithm in a scenario with a group of $N = 100$ agents. The clustering process identified 48 agent clusters. The simulation time span is set to $10[s]$ with an interval of $0.1[s]$. The PDF in Figure 5 indicates the position and its variance for the 48 agent clusters. Initially all the clusters are close together but over time high density cluster regions are formed by clusters in close vicinity to each other. The predicted mean and variance of the position allow for maneuvering around the larger agent cluster groups and high density areas.

6. CONCLUSION AND FUTURE WORK

This work proposes a Gaussian mixture based method for motion prediction of dynamic agents while accounting for formation and interaction of clusters. Agents are clustered according to their relative proximity and orientation indicating similarity in expected trajectories. The Adaptive Gaussian Sum filter propagates the state for the clusters as a mixture model and provides an estimate of their future state. Our numerical experiments indicate that the predictions from the cluster method is more efficient than those computed by standalone Gaussian Mixture models. Future work aims to update clusters based on new observations while the filter is running to improve cluster composition. and also to incorporate information about motion intent.

REFERENCES

Aoude, G., Luders, B., Joseph, J.M., Roy, N., and How, J.P. (2013). Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, 35, 51–76.

Atev, S., Miller, G., and Papanikolopoulos, N.P. (2010). Clustering of vehicle trajectories. *IEEE Trans Intell Transp Syst*, 11(3), 647–657.

Bera, A., Kim, S., Randhavane, T., Pratapa, S., and Manocha, D. (2016). Glmp-realtime pedestrian path prediction using global and local movement patterns. In *IEEE IROS 2016*, 5528–5535.

Bian, J., Tian, D., Tang, Y., and Tao, D. (2018). A survey on trajectory clustering analysis. *ArXiv*, abs/1802.06971.

Brown, A. and Rogers, J. (2017). A sampling-based probabilistic path planner for multirotor air vehicles in cluttered environments. *Proc Inst Mech Eng*, 231(1), 143–162.

Chen, J., Wang, R., Liu, L., and Song, J. (2011). Clustering of trajectories based on hausdorff distance. In *ICEIC 2011 - Int. Conf. Electron. Inf. Commun.*, 1940–1944.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, 226–231.

Fridovich-Keil, D., Bajcsy, A., Fisac, J.F., Herbert, S.L., Wang, S., Dragan, A.D., and Tomlin, C.J. (2020). Confidence-aware motion prediction for real-time collision avoidance. *Int J Rob Res*, 39(2-3), 250–265.

Helbing, D. and Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, 51(5), 4282–4286.

Hu, Y., Chen, H., Lou, J.g., and Li, J. (2007). Distributed density estimation using non-parametric statistics. In *ICDCS*, 28–28.

Jensen, C.S., Lin, D., and Ooi, B.C. (2007). Continuous clustering of moving objects. *IEEE Trans Knowl Data Eng*, 19(9), 1161–1174.

Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., and Shen, H.T. (2010). Discovery of convoys in trajectory databases. *CoRR*, abs/1002.0963.

Li, Y., Han, J., and Yang, J. (2004). Clustering moving objects. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 617–622.

Liu, S.B., Roehm, H., Heinzemann, C., Lütkebohle, I., Oehlerking, J., and Althoff, M. (2017). Provably safe motion of mobile robots in human environments. In *IEEE Robot. Autom. Lett.*, 1351–1357.

Park, C., Ondřej, J., Gilbert, M., Freeman, K., and O'Sullivan, C. (2016). Hi robot: Human intention-aware robot planning for safe and efficient navigation in crowds. In *IEEE IROS 2016*, 3320–3326.

Park, J.S., Park, C., and Manocha, D. (2019). I-planner: Intention-aware motion planning using learning-based human motion prediction. *Int J Rob Res*, 38(1), 23–39.

Patterson, A., Lakshmanan, A., and Hovakimyan, N. (2019). Intent-aware probabilistic trajectory estimation for collision prediction with uncertainty quantification. In *IEEE CDC 2019*, 3827–3832.

Schulz, J., Hubmann, C., Löchner, J., and Burschka, D. (2018). Interaction-aware probabilistic behavior prediction in urban environments. In *IEEE IROS 2018*, 3999–4006.

Terejanu, G., Singla, P., Singh, T., and Scott, P.D. (2011). Adaptive gaussian sum filter for nonlinear bayesian estimation. *IEEE Trans. Automat. Contr.*, 56(9), 2151–2156.

Trautman, P. and Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. In *IEEE IROS 2010*, 797–803.

Yokoyama, N. (2018). Decentralized conflict detection and resolution using intent-based probabilistic trajectory prediction. In *AIAA GNC 2018*, 1857.

Zheng, T., Han, Q., and Lin, H. (2021). PDE-based dynamic density estimation for large-scale agent systems. *IEEE Control Syst. Lett.*, 5(2), 541–546.

Škovierová, J., Vobecký, A., Uller, M., Škoviera, R., and Hlavac, V. (2018). Motion prediction influence on the pedestrian intention estimation near a zebra crossing. 341–348. doi:10.5220/0006694403410348.