Human Presence Probability Map (HPP): a Probability propagation based on Human Flow Grid

Jacques Saraydaryan, Fabrice Jumel, Olivier Simonin

Human Presence Probability Map (HPP): a Probability propagation based on Human Flow Grid

Jacques Saraydaryan[1,3], Fabrice Jumel[1,3] and Olivier Simonin[2,3]

[1]CPE Lyon, [2] INSA Lyon, [3] CITI Lab., INSA-INRIA Chroma team,

Abstract. Personal assistance, delivery services, and crowd navigation through robots fleet are complex activities that involve human-robot interaction and fleet coordination. Human location estimation is one of the key factors in assisting robots in their tasks. This paper proposes an efficient process for propagating human presence probability based on partial observation of humans by the robot fleet. This process provides real-time information about the most probable region on the map where humans can be found.
We propose a new problem representation allowing us to efficiently parallelize the propagation. To deal with the learned model and the real time robot observations, we propose to include a gaussian rotation probability process (VonMises [11]) combined with the previous learned observation to adapt the propagation. A set of experiments has been conduced with simulated environments that include real data allowing us to evaluate the model and to compare with the standard approach.

1 INTRODUCTION

Building robotic systems that can interact with humans remains a challenging task today. Knowledge about human behavior is crucial to develop human-acceptable and safe solutions. Navigating in populated environments or serving humans through robot fleets requires to understand human motion.

In this paper, we propose a scalable method to predict human motions in a closed environment from the observations colleted by a fleet of robots along the time. According to [14] our approach aims at computing human presence probability at middle term ($> 10s$). Indeed, middle term prediction is particularly important for robot fleet tasks achievements, where a robot could benefit of other robots observation to adapt its behavior and decision. In other words, we build a Human Presence Probability Map from robot observations.

We build our approach over an existing discrete human orientation probability map [5], called Flow Grid Map, similar to [9,19]. Then we are able to propagate online human observation information which is exploited by robots to navigate efficiently and to find humans in order to serve them.

The figure 1 shows an example of Human presence probability propagation on an entire closed environment. Here, an existing flow grid is used to help the

propagation. Each time a robot detects a human, the algorithm begins to propagate the information. The cumulative information through all robot observations results in a map highlighting the probability to meet pedestrian.

This paper is organized as follows: Section 2 describes related works in human flow modeling and human motion trajectory prediction. The section 3 explains Human flow grid models used as computation baseline. The section 4 describes our human presence probability propagation computation and associated optimizations. Finally, the section 5 shows experiments and results.

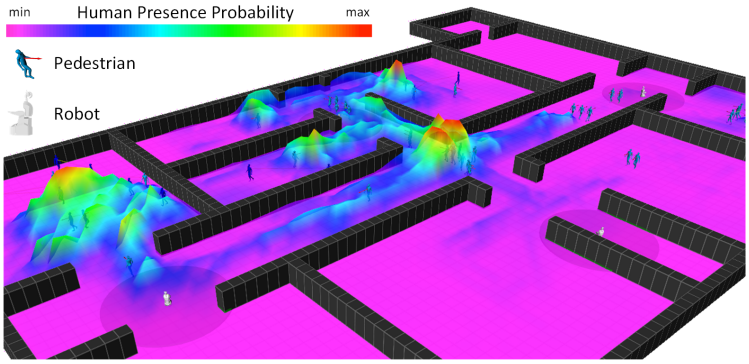

Fig. 1: HPP Map representation on closed environment

2 RELATED WORK

Human motion prediction get a growing interest recent years. Human motion prediction could be divided into two different objectives: short term prediction $(1-2s)$ and long term prediction (up to 20s) [14]. Moreover Rudenko and al. [14] propose to distinguish different kind of modeling approaches. **Physics-based modeling** aims at simulating a set of explicitly defined dynamics equation that follow a physics-inspired model. Such methods predicts human motion after sensing the environment and are mostly used in short term prediction (e.g [7]). **Planning based** methods are more focused on long-term motion. After sensing the environment, hypothesis are used to estimate path used by human to reach their goals. Finally, **Pattern-base methods** intents to learn motion model after sensing the environment. This kind of methods (sequential methods) could learn conditional models over time and recursively apply learned transitional function for inference or model the distribution over full trajectories. In this paper we particularly focus on Pattern-based approaches and especially sequential methods. In such approaches conditional models are learned from observation and mainly used Markov models. Some works intends to model local transition such as probabilities of transition between cells on a grid map [5,12,8]. Other methods

use Recurrent Neural Networks (RNN) or Long Short-term Memory (LSTM) to learn sequence of motions. Such recent methods gives good result especially on short therm prediction [20], nevertheless lots of data is needed for training such network and computed models are sensitive to motion habits changes. Other interesting approaches compute clusters on observed trajectories to predict human motion [4,21]. Despite very good prediction results, these methods suffers of same limitation as RNN or LSTM approaches. Moreover, most these methods needs either a complete trajectory monitoring (e.g start and goal position of pedestrian) or a capacity to re-identify pedestrian. Such constraints do not fit our current hypotheses of partial pedestrian observation thought a robot fleet.

As presented in section 1, the human motion prediction can help robot to compute a better navigation (socially acceptable, reducing travel time) or increase the service to human. With partial human motion observation, we need to inform the robots about human motion with learned trend and with recent observation. Furthermore, the more the map used for navigation will be informed, the more the service to human efficiency will increase. Such information is precious to help the robots to navigate [3,15]. This kind of hypotheses reduces the choose of modeling approaches. We choose to improve methods modeling transitions between cells with an efficient probability propagation process. We extends such methods [5,12,8] with the help of both learned trends and recent observation to deal with our context constraints.

3 Flow grid reminder

The authors in [5] defined a map reflecting the probability to meet a human with a certain direction. This map is computed as follows.
In each cell $c_{x,y}$, we discretize the possible flow directions by a set of K directions, and we note $k_i \in K$ each direction [1]. For instance, we can set $K = \{North, NorthEast, ..., South\}$. We note Z^t the set of observation performed by all the robots up to time t. An observation, in a cell $c_{x,y}$, consists in identifying a human direction, eventually none, and its duration. By hypothesis, only one human can occupy a cell at a given time. In practice, we consider cell sizes from $0,25m^2$ to $1m^2$.
Let note $R = r_1, r_2, ..., r_n$ the set of robots. We note $t_{c_{x,y}(r)}$ the sum of the durations of all observation (human observed or not) performed by the robot r on cell $c_{x,y}$. $t_{c_{x,y,k}(r)}$ is the sum of durations of the observation of a human moving in direction k in cell $c_{x,y}$.
We set $M_{flow} = \forall x \forall y \forall k (M_{c_{x,y,k}})$ the grid of human motion likelihood in every direction k of each cell $c_{x,y}$, called **Flow grid**. By considering that human flows are stationary processes, each cell value is computed (updated) as follows:

$$M_{c_{x,y,k}}(Z^t) = \frac{\sum_{n=1}^{|R|} t_{c_{x,y,k}}(r_n)}{\sum_{n=1}^{|R|} t_{c_{x,y}}(r_n)} \tag{1}$$

[1] When a robot detects a human in x, y pose with θ orientation $z_{x,y,\theta}$, we approximate the orientation θ to a discrete orientation k.

To deal with non stationary processes a forgetting factor could be added as is was done in [15]. By extension, the probability to meet a pedestrian in a given cell is defined by:

$$M_{pres_{c_{x,y}}}(Z^t) = \sum_{k \in K} M_{c_{x,y,k}}(Z^t) \tag{2}$$

The Flow grid could be extended if the probabilities is related to some hour of the day and day of a week as its was suggested in [19]. Thus, the likelihood to meet a human in a given orientation k is noted $M^{t_{day}}_{c_{x,y},k}$ where t_{day} represents the likelihood at a given day/week time (e.g Monday 10h12).

4 Human Presence Probability Map (HPP Map)

4.1 Basic Computation

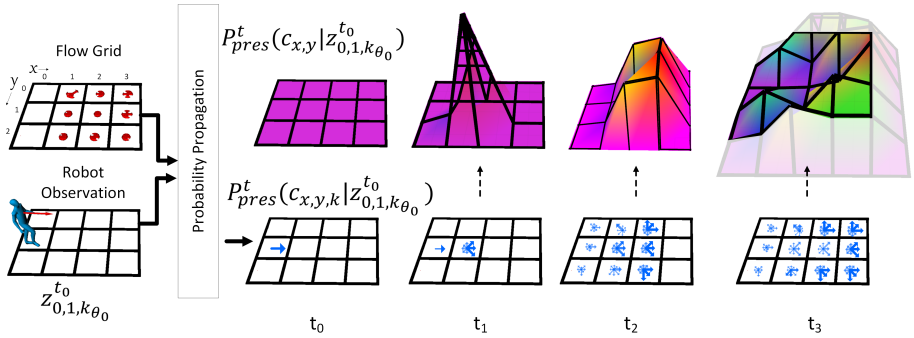

Fig. 2: Example of human presence probability propagation

Once we are able to compute information about human flow, as it was explained in section 3, we have to provide a method to propagate pedestrian presence probability when these ones have been detected by robots.

To do so, we assume that this likelihood $M_{c_{x,y},k}$ gives us a transition probability between the current cell $c_{x,y}$ and the neighbor cell targeted by the given orientation k. E.g if k means north, an current cell is $c_{1,1}$, $M_{c_{1,1},North} \equiv P(c_{1,1} \rightarrow c_{0,1})$.

Without additional information finding the most likelihood human presence on a given direction could be expressed such as: $argmax(M_{c_{x,y},k})$.
This information only takes into account human presence flow model and not recent/current human observation. In order to predict the human position in next seconds and more, we have to combine pre-computed human flow grid and current set of observation.

A common probability propagation could be expressed as follows:
For each neighbor cells around the discrete observation z^t_{x,y,k_θ}, let note $P^{t+1}_{pres}(c_{x_1,y_1,k})$

the probability to meet the previously observed human on the cell c_{x_1,y_1} at $t+1$ with the orientation k.

$$P^{t+1}_{pres}(c_{x_1,y_1,k}|z^t_{x_0,y_0,k_\theta}) = \frac{M_{c_{x_1,y_1},k}}{\sum_{i=1}^{||K||} M_{c_{x_1,y_1},k_i}} \times \alpha \times P^t_{pres}(c_{x_0,y_0,k_c}|z^t_{x_0,y_0,k_\theta}) \quad (3)$$

where k_c is the orientation pointing to the current cell c_{x_1,y_1}, k_θ is the approximated orientation θ of the observation, $P^t_{pres}(c_{x_0,y_0,k_c}|z_{x_0,y_0,k^t_\theta}) = 1$ if $k_\theta = k_c$, 0 otherwise and $\alpha \in [0;1]$ represents a stationary coefficient.

The figure 2 illustrates this principle. On the left we find input data, the learnt flow grid and a robot observation. The probability propagation process is represented from t_0 to t_3. On the bottom, the likelihood of pedestrian presence over the time is represented. Each blue arrow on each cell defines the probability to meet human in a given direction. The total probability to meet a human on each cell is represented on the top.

For a set of observation occurring at different times, noted $Z^{1:t}$, each cell accumulates the propagated probability of each observation:

$$P^t_{pres}(c_{x,y,k}|Z^{1:t}) = \frac{1}{||Z^{1:t}||} \sum_{i=1}^{||Z^{1:t}||} P^t_{pres}(c_{x,y,k}|z^{t_i}_{x_i,y_i,k_{\theta_i}}) \quad (4)$$

The resulted matrix representing the probability to meet human moving in a targeted direction is called Human Presence Probability Map (**HPP Map**)

4.2 Scalability and Propagation Optimization

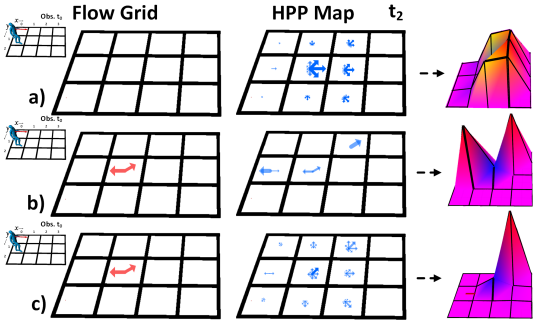

Fig. 3: Von Mises propagation optimisation: a) no information is held by the flow grid map, Von Misies coeffient are used to propagate, b) flow grid held information, propagation only used flow grid, c) both flow grid and Von Mises Coeff are used

To make easier the parallelization of the computation, we propose to reformulate the problem as follows: On one side, at a time t, a cell get the sum of cumulative

probability from a set of observation $Z^{1:t}$. On the other side, the cell contributes to the probability propagation to neighbors cells. We propose to formulate the probability of presence of a human in a given cell as follows:

$$P^t_{pres}(c_{x,y,k}|Z^{1:t}) = \frac{\alpha \times M_{c_{x,y,k}}}{\sum_{i=1}^{||K||} M_{c_{x,y,k_i}}} \times \sum_{j=1}^{||N(c_{x,y})||} P^{t-1}_{pres}(c_{x_j,y_j},k_c|Z^{1:t}) + (1-\alpha) \times P^{t-1}_{pres}(c_{x,y,k}|Z^{1:t}) \quad (5)$$

where $||N(c_{x,y})||$ is the cardinality of neighbourhood of $c_{x,y}$, $\sum_{j=1}^{||N(c_{x,y})||} P^{t-1}_{pres}(c_{x_i,y_i},k_c|Z^{1:t})$ represents the probability propagation of each neighbors cells and $(1-\alpha) \times P^{t-1}_{pres}(c_{x,y,k}|Z^{1:t})$ the previous cell value minus the current cell propagation effort to neighbor cells.

Such reformulation allows massive parallelization. At t each HPP Map cells could computes its $t+1$ value without waiting computation from neighbors. This property allows a GPU implementation of the HPP Map computation.

Moreover, some flow grid cell could be uninformed, meaning that no observation has been made in flowgrid map. In such situation, we assume that human motion follows a Gaussian orientation change model as described by Von Mises equation such as: $VonMises(\theta_k, \theta_{k_{ref}}) = \frac{e^{Ccos(\theta_k - \theta_{k_{ref}})}}{2\pi I_0(C)}$ where $\frac{1}{C} = \sigma^2$ and θ_k and $\theta_{k_{ref}}$ represents the angle of a given orientation (e.g if K means North, $\theta_k = \frac{\pi}{2}$) respectively for a given orientation and a referent orientation. I_0 is the modified Bessel function of order 0.

Thus, for uninformed flow grid cell, the probability propagation could be defined as:

$$P^{t+1}_{pres}(c_{x_1,y_1,k}|z^t_{x_0,y_0,\theta}) = VonMises(\theta_k, \theta_{k_c}) \times \alpha \times P^t_{pres}(c_{x_0,y_0,k_c}|z^t_{x_0,y_0,k_\theta}) \quad (6)$$

The figure 3.a shows an example of application.

The flowgrid maps give a trend of human motion. In some situation, some observation not fit computed model. E.g if an observation was made from a cell to a given direction, if the targeted cell contains high probabilities to go back to the current cell, the system would not take into account the characteristic of the current observation. To avoid such behavior, we propose to use again the Von Mises function to combine current observation motion and human motion trends. This approach will temper probability propagation by combining the current observation and the learnt flow grid map. The resulting value is expressed as follows:

$$P^{t+1}_{pres}(c_{x_1,y_1,k}|z^t_{x_0,y_0,\theta}) = \frac{VonMises(\theta_k, \theta_{k_c}).M_{c_{x,y,k}}}{\sum_{i=1}^{||K||} M_{c_{x,y,k_i}}} \times \alpha \times P^t_{pres}(c_{x_0,y_0,k_c}|z^t_{x_0,y_0,k_\theta}) \quad (7)$$

The figure 3 shows an example of propagation using such equation. In the figure 3.b situation only flow grid information is used to propagate probability. In such case half of probability is send back to the cell where the observation was made. Such behavior means that the human has 50 per cent to half turn. In the figure 3.c case, we apply Von Mises coefficients to prioritize ongoing moves, reflecting a more realistic propagation.

5 Experimentation

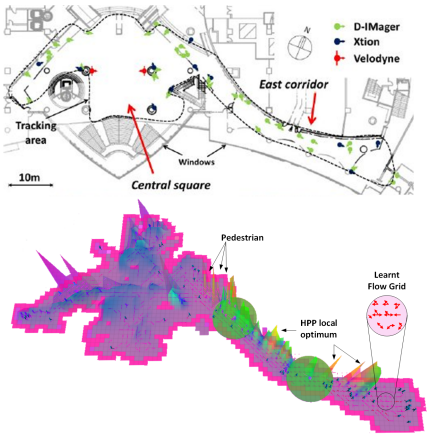

Fig. 4: Real ATC scenario. On the top, the original configuration scenario as mentioned in [2]. On the bottom, the scenario played with 3 robots following Hamiltonian path, flow grid learning and HPP computation

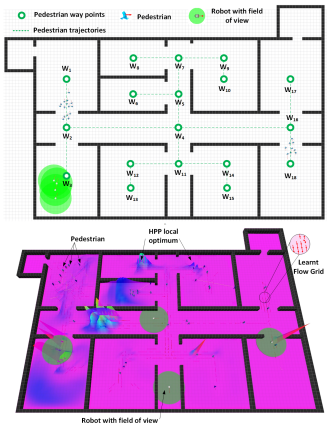

Fig. 5: Museum Scenario (Artificial Scenario). On the top, the scenario map and the set of way points following by 2 pedestrian groups. On the bottom, the scenario played with 4 robots following Hamiltonian path, flow grid learning and HPP computation

Our first series of experiments was made on a simulated environment using the ROS version of Pedsim simulator [1]. This experiment involved 50 pedestrians walking through pre-defined trajectories (Figure 5). The environment is composed of several corridors and rooms. In this experiment case, we want to highlight the performance improvement of computing Human Presence Probability propagation with a standard implementation with python and numpy, and a GPU implementation using Numba [10]. For experimentation, we use the flow grid described in our previous worked [5]. In this scenario, 4 robots allowing wide-ranging observations (360° field of view with a range of 3 meters) navigate across the environment according Hamiltonian path [13] (optimising the space coverage). Thus the robot fleet covers the environment and updates the human

presence (z_{x,y,k_θ^t}) when a pedestrian is met (in the robot field of view). We assumed that probabilities coming on map walls are removed. This experiment was conducted during 800 seconds. The experiment was made on a laptop i7, 9th gen., 16Go RAM, graphic card NVIDIA Quatro T1000. With an average processing time of $1.632s$ (to propagate the HPP of the entire environment at each processing step) for the standard implementation and $0.052s$ for the GPU implementation, this gain of more than 30 times make the usage of the HPP map computing online realistic.

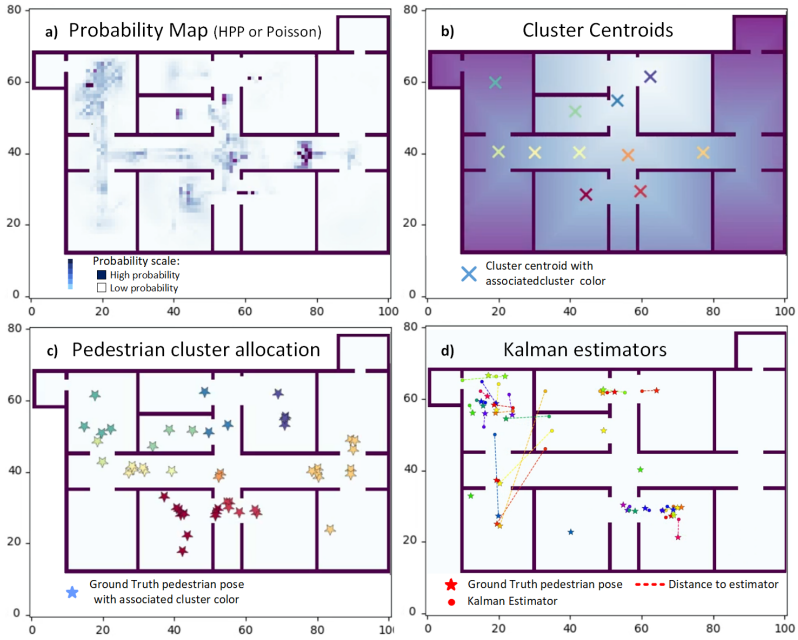

Fig. 6: Pedestrian position prediction evaluation steps

During this scenario, most probable regions to meet people are clustered (HPP clusters) and human position ground truth is used to evaluate the distance between real human position and HPP clusters. In other words, the lower the average distance is, the more the pedestrian estimation is accurate. The figure 6.b shows the HPP Cluster centroids. To build such clusters, we first filter HPP Map (figure 6.a) data that are above a threshold. Here we assume that probabilities below such threshold are not sufficient to be representative of Human position estimation. A DBscan [16] clustering algorithm is then applied and resulted cluster centroids are displayed by a colored cross in the figure 6.b). At the end, tracked pedestrian are affected to the closest cluster centroid (wave front propagation from the targeted cluster center) and distance between pedestrian position and cluster centroid is measured. The error measure could be expressed

by $error_{h_i} = \underset{clust \in DBscan(HPPMap)}{\arg\min} \Big(dist(clust, h_i) \Big)$, where $clust$ is a cluster centroid and h_i the ground truth of a pedestrian pose.
We compare our results obtained in such scenario with two other approaches:

- one long term design approach: human affordance map [18] aims at building a statistical representation of human presence (only human presence per cell is considered in such approach).
- one short term design approach: Kalman prediction, where humans are tracked through well known Kalman filter [6].

The human affordance map uses robot observation to update the probability to meet human on each map cell during the time. The human affordance map is updated all experimentation long. As it is done for HPP map, at each evaluation step, a DBscan cluster algorithm distinguishes a set of most probable human presence area based on the human affordance map. Then each observed human is affected to the closest cluster. The distance between human and its associated cluster centroid is considered to evaluate the prediction precision (figure 6.c).

The Kalman prediction is also used to predict pedestrian pose. When robot meet a pedestrian, a Kalman filter is created and updated as long as the associated pedestrian is in the field of view of a robot. Otherwise the Kalman filter continue to predict the next pedestrian position. The distance between the pedestrian pose ground truth and the Kalman prediction is considered to evaluate the prediction precision (figure 6.d). All Kalman filter are set with the associated pedestrian pose, an acceleration of $0.68 m.s^2$ and an acceleration standard deviation of 0.2 based on recommendation of[17].
The evaluation process was made as follows: pedestrian detected by robot are tracked (ground truth position) during different time range (from $10s$ to the end of experiment). For each tracked pedestrian, the distance between ground truth pose and associated prediction is measured for each evaluation process:

- HPP : Distance to the closed hpp cluster
- Human affordance : Distance to the closed human affordance cluster
- Kalman prediction : Distance to the associated Kalman filter prediction pose.

The results of the average distance of all ground truth pedestrian poses and HPP cluster centroids are available in the Figure 7.

In the Figure 7.a, the average distance is measure during the time. Each colors refers to different pedestrian track duration. For instance, the orange curve refers to a track during of $10s$ meaning that over $10s$ of a pedestrian tracking, this one is no more tracked and its ground truth position is forgotten. We can notice that the average distance decrease over the time. This observation is due to the enrichment of the flow grid map, increasing the accuracy of the HPP. In the figure 7.b the cumulative average distance is computed. It comes as no surprise that lower pedestrian tracking duration implies best pedestrian prediction. We can notice that tracking pedestrian during $30s$ (brown curve) or tracking over the whole experimentation ($800s$, blue curve) results of no significant difference compare to duration of $10s$ or $30s$.

The results of the average distances (average of all ground truth pedestrian poses and cluster centroids/estimators) of each processes are available in the table 1. In this table, we compare two versions on HPP, one with the VonMises propagation model, and one without. The capacity of the VonMises propagation model to balance current pedestrian observation/direction and the learnt flow gives better results on all different pedestrian tracking duration. The HPP with VonMises get best results on all pedestrian tracking durations except for $800s$ duration. In this case, Human affordance coupled with the flow grid already learnt get better results. This can be explain by the fact that the more the tracking duration is the more the pedestrian as a chance to not being met by a robot and so the static statistical Human probability representation (human affordance map) get best results. We also highlight the influence of the flow grid, both learning flow grid from the start and previously learnt flow grid at the exprimentation start have been played. We can notice that concerning the Human affordance map or HPP Map results are significantly better when flow grid is already learnt.

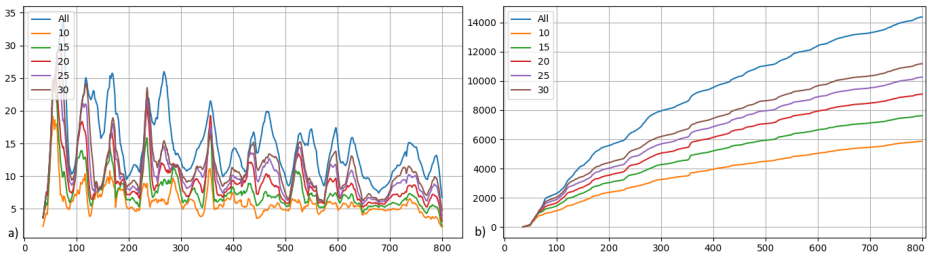

Fig. 7: Average distance from ground truth and HPP clusters during the time. The figure a shows average distance from HHP cluster using a sliding windows of 20. The figure b represents the cumulative average error distance over the time.

The second set of experiments was made on a subset of the ATC pedestrian tracking data set [2]. This dataset is the result of tracking people on an ATC shopping center in Japan over about $900m^2$ between October 24, 2012 and November 29, 2013. We extract a subset of this dataset (data of the 2012/11/14). The interaction between robot's moves and the crowd is considered by integrating real data into the pedsim simulator:

- Extract consecutive series of individual pedestrian position
- Sample individual pedestrian position
- Convert samples pose into way points
- Create pedestrian at the apparition time, following its associated way points

Between way points, each pedestrian is driven by Pedsim local planner (repulsive and attractive potential forces) and thus avoid collision between other

simulated pedestrian or robot. The same evaluation process used in the first experiment is applied. The results of the average distances (average of all ground truth pedestrian poses and cluster centroids/estimators) of each processes of the ATC scenario are available in the table 1.

The results show the same trends found the first experiment. HPP Map with VonMises get the best results no matter the pedestrian tracking duration. Results became also better when an already learnt flow grid is used. In this set of experiment, Human affordance process do not overcome the HPP Map method for long pedestrian tracking (e.g 800s). This result is mainly due to the fact that in the ATC dataset pedestrian appear and disappear on quite short period meaning that a tracking no more exceed 30s.

The experiments carried out are illustrated in the video here [2]. Our proposition outperforms other tested models by an average of 17% (and up to 40% in certain conditions).

	Museum Scenario						Real ATC scenario					
Tracking duration (in s.)	10	15	20	25	30	800	10	15	20	25	30	800
HPP Without VonMises	10.60	12.57	13.96	14.96	15.66	21.31	Not Tested					
HPP With VonMises	6.58	8.54	10.20	11.50	12.53	16.12	**7.45**	**8.11**	**8.52**	**8.77**	**8.91**	**9.00**
HPP With VonMises Flow grid learnt[3]	**5.38**	**6.01**	**6.81**	**7.87**	**8.79**	13.02	**4.29**	**4.95**	**5.44**	**5.49**	**5.53**	**5.81**
Human affordance	13.46	13.86	14.01	14.04	13.96	12.99	14.63	15.06	15.84	16.44	16.55	16.55
Human affordance Flow grid learnt [3]	9.29	9.27	9.14	9.13	9.14	**9.03**	9.00	9.83	10.65	11.05	11.26	11.90
Kalman Prediction	13.01	14.83	16.582	18.05	19.11	24.01	11.69	13.67	15.21	16.16	16.57	16.63

Table 1: Average distance (in meter) of pedestrian between computed pedestrian pose estimation and ground truth from Museum scenario (artificial scenario) and Real ATC dataset

6 CONCLUSION

In this paper, we propose a new method for propagating human presence probability based on partial human observation by a robots fleet. With the help of a Flow grid Map, representing the probability to meet human in a cell in a direction, we propagate probabilities as soon as a robot observes a human. By reformulating the problem, we manage to massively parallelize computation (GPU) allowing online Human Presence Probability propagation. Thanks to the applied optimization (VonMises), our method is able to handle both learned trends and current observations. Our experimental results, conducted on two different scenarios (an artificial and a real-life dataset), demonstrate that our model outperforms existing long-term (Human Affordance Map) and short-term (Kalman estimator) human presence estimators.

[2] `https://youtu.be/H7Ly9nZKNks`

[3] For ATC Scenario 500000 records was used to learn the flow grid, the next 500000 records are used to evaluate the model

References

1. Ros packages for pedsim (pedestrian simulator) based on social force model of helbing et. al, `https://github.com/srl-freiburg/pedsimros`
2. Brscic, D., Kanda, T., Ikeda, T., Miyashita, T.: Person tracking in large public spaces using 3-d range sensors. IEEE Transactions on Human-Machine Systems (2013)
3. Fulgenzi, C., Spalanzani, A., Laugier, C.: Probabilistic motion planning among moving obstacles following typical motion patterns. In: IEEE/RSJ International Conference on Intelligent RObots and Systems (2009)
4. Han, Y., Tse, R., Campbell, M.E.: Pedestrian motion model using non-parametric trajectory clustering and discrete transition points. CoRR (2020)
5. Jumel, F., Saraydaryan, J., Simonin, O.: Mapping likelihood of encountering humans: application to path planning in crowded environment. In: The European Conference on Mobile Robotics (ECMR) (2017)
6. Kalman, R.E., Others: A new approach to linear filtering and prediction problems. Journal of basic Engineering (1960)
7. Kooij et al.: Context-based path prediction for targets with switching dynamics (2019)
8. Kucner et al.: Enabling Flow Awareness for Mobile Robots in Partially Observable Environments. IEEE Robotics Autom. Lett. (2017)
9. Kucner et al.: Probabilistic Mapping of Spatial Motion Patterns for Mobile Robots. In: Cognitive Systems Monographs (2020)
10. Lam, S.K., Pitrou, A., Seibert, S.: Numba: A llvm-based python jit compiler. In: Proceedings of the 2nd Workshop on the LLVM Compiler Infrastructure in HPC (2015)
11. Mardia, K., Jupp, P.: Directional Statistics. (2000)
12. Mellado et al.: Modelling and predicting rhythmic flow patterns in dynamic environments. In: Towards Autonomous Robotic Systems TAROS (2018)
13. Portugal, D., Rocha, R.: A survey on multi-robot patrolling algorithms. In: Technological Innovation for Sustainability (2011)
14. Rudenko et al.: Human motion trajectory prediction: a survey. The International Journal of Robotics Research (2020)
15. Saraydaryan, J., Jumel, F., Simonin, O.: Navigation in Human Flows : Planning with Adaptive Motion Grid. In: IROS Workshop CrowdNav (2018)
16. Schubert et al.: Dbscan revisited, revisited: Why and how you should (still) use dbscan. ACM Trans. Database Syst. (2017)
17. Teknomo, K.: Microscopic pedestrian flow characteristics: Development of an image processing data collection and simulation model. arXiv (2016)
18. Tipaldi, G.D., Arras, K.O.: I want my coffee hot! Learning to find people under spatio-temporal constraints. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2011)
19. Vintr et al.: Spatio-temporal Representation of Time-varying Pedestrian Flows. In: ICRA Workshop on Long-term Human Motion Prediction (2019)
20. Xie, Z., Xin, P., Dames, P.M.: Towards safe navigation through crowded dynamic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2021)
21. Zhou, B., Tang, X., Wang, X.: Learning collective crowd behaviors with dynamic pedestrian-agents. International Journal of Computer Vision (2014)